UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

ADAPTACIÓN AUTOMÁTICA DE LA DETECCIÓN DE PERSONAS A LA ESCENA

Aarón Monedero Grifo
Tutor: Álvaro García Martín
Ponente: José M. Martínez Sánchez

Julio 2017

ADAPTACIÓN AUTOMÁTICA DE LA DETECCIÓN DE PERSONAS A LA ESCENA

Aarón Monedero Grifo
Tutor: Álvaro García Martín
Ponente: José M. Martínez Sánchez

Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2017

Trabajo parcialmente financiado por el Ministerio de Economía y Competitividad del Gobierno de España bajo el proyecto TEC2014-53176-R (HAVideo) (2015-2017)

Resumen

Actualmente los sistemas de detección de personas son ampliamente utilizados en aplicaciones de videovigilancia. A pesar de haber ya diversos estudios sobre como mejorar los resultados obtenidos, esta área continua siendo de gran interés debido a su relevancia y complejidad.

Este trabajo se enfoca en cumplir la premisa de que estos sistemas de seguridad deben ser robustos para poder detectar personas en cualquier entorno. En ocasiones, los escenarios que encontramos a la hora del análisis son realmente complicados para una correcta detección. Se dan situaciones adversas reales muy variadas, tanto en interiores como en exteriores, de cambios de iluminación y sombras o personas parcialmente ocluidas por objetos.

La idea principal es utilizar el contexto de escena en cada vídeo para reentrenar el modelo y de esta forma conseguir una detección más precisa para la mayoría de los casos. El trabajo ha consistido en implementar mediante Matlab una adaptación del algoritmo ACF (Aggregate Channel Features) para la detección de personas. En primer lugar se realiza una aproximación para demostrar la idea y a continuación se desarrolla el algoritmo propuesto. Se realiza también el estudio de los parámetros variables del sistema para alcanzar los mejores resultados.

Finalmente, se ha realizado una evaluación del algoritmo con sus diferentes variaciones sobre las secuencias de referencia. Los resultados obtenidos demuestran la mejora en las detecciones para las secuencias analizadas.

Palabras clave

Detección de personas, videovigilancia, contexto, escena, adaptación, modelo, ACF

Abstract

Nowadays, people detection systems are widely used in video surveillance applications. In spite of the existence of a variety of studies about improving the obtained results, this area continues being of great interest due to its relevance and complexity.

This work is focused on accomplishing the premise that these security systems must be robust in order to be able to detect people in every scene. Occasionally, the scenerios we find at the time of the analysis are really complex for a proper detection. There might arise a number of real and adverse situations, both indoors and outdoors, of lighting and shadows alterations or people partially occluded by objects.

The main idea is using the scene context of each video to adapt the model in pursuance of getting a more accurate detection for the majority of the cases. The work has consisted in implementing through Matlab an adaptation of the ACF's algorithm (Aggregate Channel Features) for people detection, as well as in the study of the system's variable parameters to achieve the best results.

At last, an evaluation of the algorithm with its different variations about the reference sequences has been done. The obtained results demonstrate the improvement in the detections for the analyzed sequences.

Keywords

People detection, video surveillance, context, scene, adaptation, model, ACF

Agradecimientos

En primer lugar me gustaría agradecer a mi tutor, Álvaro, por darme la oportunidad de realizar este trabajo, por su paciencia y disponibilidad para ayudar siempre que haga falta.

Gracias a los compañeros que durante estos años han compartido conmigo clases, trabajo y prácticas, pero también muchos buenos momentos e historias que siempre se recordarán. Por hacer de las horas en la universidad un tiempo más agradable y demostrarme que existe esa amistad también lejos de los libros.

Quiero agradecer de igual manera a mis compañeros de piso y amigos por cada día hemos podido compartir desde que hace ya unos cuantos años pusimos pie en Madrid. Muchas gracias a también a mi grupo de amigos de Cuenca, por hacerme desconectar, por estar siempre presentes y, en definitiva, por ser esa increíble familia que uno elige. Tengo que agradecer también a mi prima Celia, que en buena parte es responsable de que yo decidiese estudiar esta carrera, por poder contar siempre con ella, porque en cualquier momento puede surgir un gran plan para pasar el día.

Muchas gracias a mis abuelos por confiar siempre en mí y a mi hermana, quien parece decidida a coger el relevo y ya pone su granito de arena en el trabajo. Pero por supuesto si hay alguien a quien tenga que dar las gracias por encima de todo es a mis padres, quienes son la principal razón por su esfuerzo y dedicación de que esté presentando este trabajo.

Porque ya es más vuestro que mío, para todos va dedicado este trabajo.

Índice general

Resumen v

Abstract vii

Agradecimientos ix

1. Introducción 1
1.1. Motivación . . . 1
1.2. Objetivos . . . 2
1.3. Organización de la memoria . . . 2

2. Estado del arte 5
2.1. Introducción a la detección de personas . . . 5
2.1.1. Arquitectura de los sistemas de detección de personas . . . 6
2.1.2. Clasificación del estado del arte de la detección de personas . . . 7
2.1.3. Fast Features Pyramids . . . 10

3. Diseño y desarrollo 17
3.1. Introducción . . . 17
3.2. Algoritmo base . . . 17
3.2.1. Diseño . . . 17
3.2.2. Desarrollo . . . 18
3.3. Algoritmo propuesto . . . 22
3.3.1. Adaptación del algoritmo a la escena . . . 22
3.3.2. Diseño Dummy . . . 22
3.3.3. Caltech Pedestrian Dataset para generar dataset de entrenamiento . . . 23

4. Evaluación 27
4.1. Introducción . . . 27
4.2. Marco de evaluación . . . 27
4.2.1. *Dataset* . . . 27
4.2.2. Métricas de evaluación . . . 28
4.3. Pruebas y resultados . . . 29
4.3.1. Pruebas y resultados del algoritmo base frente a la versión Dummy 29

4.3.2. Pruebas y Resultados del algoritmo propuesto 32

5. Conclusiones y trabajo futuro **35**
5.1. Conclusiones 35
5.2. Trabajo futuro 36

Bibliografía **38**

A. Resultados del algoritmo base frente a la versión Dummy **41**

B. Resultados del algoritmo propuesto **51**

Índice de figuras

2.1. Arquitectura de un sistema de detección de personas 6
2.2. Comportamiento de los histogramas de gradientes en imagenes remuestreadas por un factor dos. 11
2.3. Histogramas de gradientes aproximados en imágenes remuestreadas por un factor dos.[1] 13
2.4. Cálculo del canal de características escalado.[1] 14
2.5. Construcción de la pirámide rápida de características. [1] 14
2.6. MR para detección de personas en cuatro datasets. [1] 16

3.1. Visión conjunta del esquema de funcionamiento del detector ACF . . . 18
3.2. Mapa de características de los histogramas de gradientes 19
3.3. Resultado de la detección sobre un frame de ejemplo utilizando el detector base. 21
3.4. Tasa de error y falsos positivos por imagen del detector base. 21
3.5. Imágenes incorporadas al dataset de entrenamiento como ejemplos positivo (izq.) y negativo (dcha.). 23
3.6. Imágenes negativas a partir de un mismo frame con percentil 5 (dcha.) y 20 (izq.). 25

4.1. Tres imágenes positivas de ejemplo del dataset INRIA 28
4.2. Tres imágenes de ejemplo del dataset Caltech 28
4.3. Curva Precisión-Recall sobre la secuencia 'abandonedBox' utilizando la versión Dummy 30
4.4. Frame de ejemplo de la secuencia 'abandonedBox' 31
4.5. Curva Precisión-Recall sobre la secuencia 'abandonedBox' utilizando el algoritmo propuesto 32

A.1. Curva Precisión-Recall sobre la secuencia 'PETS2006' 42
A.2. Curva Precisión-Recall sobre la secuencia 'backdoor' 42
A.3. Curva Precisión-Recall sobre la secuencia 'badminton' 43
A.4. Curva Precisión-Recall sobre la secuencia 'busStation' 43
A.5. Curva Precisión-Recall sobre la secuencia 'copyMachine' 44
A.6. Curva Precisión-Recall sobre la secuencia 'cubicle' 44
A.7. Curva Precisión-Recall sobre la secuencia 'fall' 45
A.8. Curva Precisión-Recall sobre la secuencia 'office' 45

A.9. Curva Precisión-Recall sobre la secuencia 'overpass' 46
A.10. Curva Precisión-Recall sobre la secuencia 'pedestrians' 46
A.11. Curva Precisión-Recall sobre la secuencia 'peopleInShade' 47
A.12. Curva Precisión-Recall sobre la secuencia 'sidewalk' 47
A.13. Curva Precisión-Recall sobre la secuencia 'skating' 48
A.14. Curva Precisión-Recall sobre la secuencia 'sofa' 48
A.15. Curva Precisión-Recall sobre la secuencia 'tramstop' 49
A.16. Curva Precisión-Recall sobre la secuencia 'wetSnow' 49
A.17. Curva Precisión-Recall sobre la secuencia 'winterDriveway' 50
A.18. Curva Precisión-Recall sobre la secuencia 'zoomInZoomOut' 50

B.1. Curva Precisión-Recall sobre la secuencia 'PETS2006' 52
B.2. Curva Precisión-Recall sobre la secuencia 'backdoor' 52
B.3. Curva Precisión-Recall sobre la secuencia 'badminton' 53
B.4. Curva Precisión-Recall sobre la secuencia 'busStation' 53
B.5. Curva Precisión-Recall sobre la secuencia 'copyMachine' 54
B.6. Curva Precisión-Recall sobre la secuencia 'cubicle' 54
B.7. Curva Precisión-Recall sobre la secuencia 'fall' 55
B.8. Curva Precisión-Recall sobre la secuencia 'office' 55
B.9. Curva Precisión-Recall sobre la secuencia 'overpass' 56
B.10. Curva Precisión-Recall sobre la secuencia 'pedestrians' 56
B.11. Curva Precisión-Recall sobre la secuencia 'peopleInShade' 57
B.12. Curva Precisión-Recall sobre la secuencia 'sidewalk' 57
B.13. Curva Precisión-Recall sobre la secuencia 'skating' 58
B.14. Curva Precisión-Recall sobre la secuencia 'sofa' 58
B.15. Curva Precisión-Recall sobre la secuencia 'tramstop' 59
B.16. Curva Precisión-Recall sobre la secuencia 'wetSnow' 59
B.17. Curva Precisión-Recall sobre la secuencia 'winterDriveway' 60
B.18. Curva Precisión-Recall sobre la secuencia 'zoomInZoomOut' 60

Índice de tablas

4.1. Evaluación Ground Truth del algoritmo Dummy 31
4.2. Evaluación Ground Truth del algoritmo propuesto 33

Capítulo 1

Introducción

1.1. Motivación

Durante los últimos años ha habido grandes avances y una evolución constante en lo que al procesamiento de imagen y vídeo digital se refiere. Se han realizado grandes esfuerzos y progresos en la investigación en este campo debido a su relevancia en áreas como la vídeo-seguridad, que se ha vuelto imprescindible en la sociedad actual. Por esto, las líneas de investigación están orientadas a innovar o desarrollar las técnicas de análisis existentes con el fin de brindar a las personas seguridad diaria.

Dentro del campo de investigación de procesado de imagen y vídeo, y más concretamente de la videovigilancia, nos encontramos con múltiples algoritmos para la detección de objetos, caídas o sucesos. La detección automática de personas forma parte de este grupo y cuenta con aplicaciones importantes en seguridad o inteligencia artificial, como prueba tenemos la gran cantidad de detectores de personas que existen actualmente.

A pesar de la complejidad en la detección de personas, encontramos algunos algoritmos que presentan resultados óptimos. Sin embargo, cuando se plantea la dificultad de evaluar la detección sobre escenarios complejos el rendimiento que ofrecen se ve afectado notablemente. La variabilidad de perspectivas, posturas, distancia de las personas a la cámara o interacciones entre personas y objetos son algunos de los factores que diferencian estos entornos.

Por todo esto, la motivación del trabajo es conseguir una solución efectiva para obtener buenos resultados en situaciones complejas. La vía utilizada será contar con la información de contexto de la escena para adaptar el detector de personas y evaluarlo sobre las diferentes secuencias de vídeo.

1.2. Objetivos

Establecida la motivación del trabajo, establecemos los objetivos partiendo desde propósitos parciales que finalmente permitirán alcanzar el objetivo global de una manera progresiva y adecuada. El desglose de los objetivos es el siguiente:

1. **Estudio detallado del estado del arte.**
   - Conocer en que consiste la detección de personas, analizando algunos de los algoritmos y enfoques de los detectores para llevar a cabo el propósito requerido.
   - Análisis en profundidad del detector ACF [1], estudiando su funcionamiento y particularidades.
2. **Aprendizaje de herramientas y bibliotecas.**

   Utilizar las herramientas y bibliotecas necesarias para el correcto funcionamiento de este trabajo: Matlab [2]
3. **Desarrollo de la adaptación del algoritmo de detección de personas ACF.**

   Implementación de un detector que utilice información de contexto de la escena, desarrollando el algoritmo seleccionado en varias etapas buscando obtener una solución práctica y realista que ofrezca buenos resultados.
4. **Evaluar y analizar los resultados obtenidos.**

   Comparar la eficacia del sistema implementado en las diferentes secuencias de vídeos para conocer sus ventajas, limitaciones y posibles mejoras.

1.3. Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo 1. Introducción y motivación del proyecto
- Capítulo 2. Estado del arte de la detección de personas. Estudio de las características y peculiaridades del algoritmo elegido.
- Capítulo 3. Descripción de la adaptación y programación del sistema de detección de personas utilizando contexto de la escena.

- Capítulo 4. Estudio del criterio de evaluación y análisis comparativo de los resultados obtenidos para el detector base y nuestro sistema de detección utilizando información de contexto.
- Capítulo 5. Conclusiones obtenidas tras el análisis de resultados del trabajo presentado. Problemas pendientes, posibles mejores y trabajo futuro.
- Bibliografía.

Capítulo 2

Estado del arte

Al igual que muchas otras tecnologías, la detección mediante visión artificial surge de manera paulatina junto a las mejoras en otras tecnologías como los sistemas de captación o los ordenadores. Los primeros trabajos que trataban sobre el tema comienzan a aparecer en las décadas de los 50 y 60 con trabajos importantes como el realizado por Roberts [3] en el que demuestra la posibilidad de extraer una descripción matemática de los objetos contenidos en una imagen digitalizada.

Desde hace ya unos años, se viene dando un continuo desarrollo de aplicaciones y algoritmos que hace de los sistemas de visión artificial, y concretamente de la detección de personas, una disciplina en constante evolución que despierta un gran interés en la comunidad científica.

Actualmente existen múltiples trabajos científicos centrados en la detección de personas. En este capítulo se presenta al lector el área de estudio en la que se enmarca el trabajo, comentando las técnicas existentes, y estará dividido en dos secciones. En la primera de ellas se lleva a cabo un análisis de la detección de personas, comentando las fases en las que se divide la detección. Posteriormente en la segunda sección el objetivo es profundizar en el estado del arte del detector de personas empleado en este trabajo, así como de los algoritmos que sirven de punto de partida para dicho detector.

2.1. Introducción a la detección de personas

La detección de personas consiste principalmente en el diseño y entrenamiento de un modelo de persona basado en parámetros característicos de la misma como sus dimensiones,silueta o movimiento, para después ajustar este modelo a los posibles objetos de la escena y determinar si son personas o no. La detección de personas en

secuencias de vídeo es una de las tareas más complejas debido a la dificultad para definir un modelo genérico de persona. Únicamente los objetos candidatos que se ajusten al modelo serán clasificados como personas.

2.1.1. Arquitectura de los sistemas de detección de personas

El detector de personas es el sistema que se encarga de determinar si en una imagen existen personas. En caso afirmativo se deberá proporcionar el tamaño y posición del individuo clasificado sobre la imagen. Según [4] la arquitectura del detector de personas se puede entender desde el estudio de sus diferentes etapas:

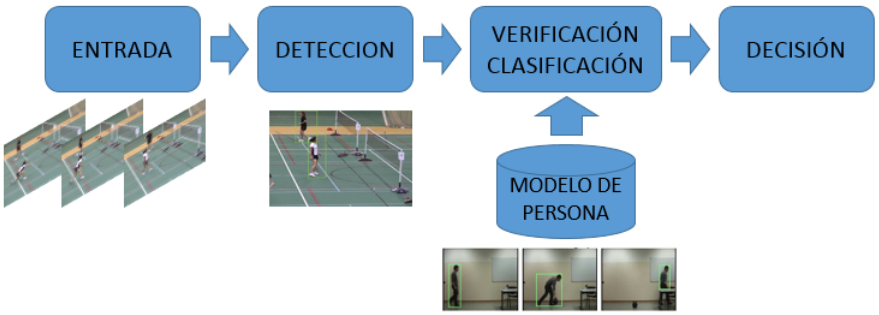

Figura 2.1: Arquitectura de un sistema de detección de personas

1. **Entrada**: Existen multitud de formatos posibles que determinan la información disponible a la entrada del detector. Respecto a la visión por ordenador, la unidad básica de entrada es la imagen o secuencia de frames en el caso del procesamiento de vídeo.

2. **Detección**: En esta fase el objetivo es generar un conjunto de regiones candidatas (*regions of interest* o ROI) que serán las que se clasificarán en la siguiente fase. Se trata de una tarea crítica que determinará en la detección global factores como la velocidad de procesamiento, los resultados de la detección o la robustez a la variaciones de la escena. Se trata de facilitar la clasificación, reduciendo lo máximo posible las zonas de fondo pero también evitando hacer una selección tan estricta que suponga la pérdida de candidatos.

3. **Modelo de persona**: El modelo definirá las características y reglas que los objetos de la imagen deben cumplir para ser considerados personas. Junto con la fase anterior de detección, es una tarea crítica en la detección de personas

que será decisiva en la fortaleza a variaciones de pose, oclusiones o velocidad de procesamiento.

4. **Verificación o Clasificación**: Esta tarea compara modelos entrenados y el modelo generado a partir de la imagen o secuencia. El objetivo es determinar si las regiones obtenidas en la fase de detección son personas o no lo son, minimizando el número de falsos positivos (clasificar como persona una ROI que no lo es) y los falsos negativos (no clasificar una persona presente en la imagen). En cuanto a la verificación, se realiza tras la clasificación para comprobar los resultados obtenidos y filtrar falsos positivos

5. **Decisión**: De acuerdo a la similitud calculada en la etapa previa, se debe tomar la decisión final de si se trata de una persona o no. En función de la aplicación puede no ser una tarea binaria y expresarse como una confianza o probabilidad de ser una persona.

2.1.2. Clasificación del estado del arte de la detección de personas

Existen numerosos criterios para clasificar los algoritmos de detección de personas, como las técnicas empleadas (extracción de frente y fondo, estimación de movimiento, etc.), el uso de información 2D o 3D o si los sensores son estacionarios o móviles. Tal como hemos visto en la sección anterior, las dos labores críticas son la detección del objeto y el modelo de persona empleado, que determinarán en gran medida el resultado en la detección global.

2.1.2.1. Detección de objetos

En [4] vemos que son dos los enfoques para la detección de objetos. Se podrán clasificar los detectores en aquellos basados en segmentación frente-fondo, los que se fundamentan en la búsqueda exhaustiva y, existen también, los que tratan de combinar ambas perspectivas. El resultado final será la localización y dimensionamiento de los objetos candidatos a ser personas.

- **Segmentación**:

  Actualmente algunos de los algoritmos del estado del arte [5] incorporan un primer módulo de segmentación en su tarea de detección de personas. Este paso se basa en distinguir entre el primer plano en movimiento (foreground) y el fondo (background) para conseguir agrupar los píxeles de los objetos en movimiento y aislarlos del fondo. La manera de realizar la segmentación es extrayendo el fondo a partir de un modelo preciso de fondo. Existen también algoritmos que

usan la información de color para realizar la segmentación [6]. El background normalmente contiene objetos estáticos pero puede presentar cambios a lo largo del tiempo, por lo que la imagen de fondo estará formada por píxeles estáticos y dinámicos.

En relación con la detección de personas, el uso de la segmentación genera directamente los objetos candidatos a ser personas. Por ello, la tarea de clasificación así como el modelo de persona podrá simplificarse para tener un coste computacional mucho menor. Sin embargo, esta dependencia de la segmentación puede afectar al rendimiento limitando las detecciones e incluyendo un mayor número de falsos positivos, además de ser muy complicado obtener una buena segmentación en escenarios complejos.

- **Búsqueda exhaustiva**:

  La técnica de búsqueda exhaustiva consiste en escanear la imagen en su totalidad buscando similitudes con el modelo de persona empleado, este proceso se realiza a múltiples escalas y posiciones. Tras la búsqueda se obtendrá un mapa de confianza muy denso en el cual, para llegar a detecciones individuales, se deben buscar los máximos locales y aplicar una supresión de no-máximos. El método de búsqueda exhaustiva es el más utilizado por los algoritmos de detección de personas y pueden destacarse dos enfoques diferentes.

  Por un lado existen los algoritmos basados en ventana deslizante, que obtienen el mapa de densidad evaluando con diferentes ventanas de detección con un clasificador. El tamaño de las ventanas suele guardar una razón de aspecto similar a la del cuerpo humano, como por ejemplo en [7] donde los autores utilizan una ventana de 64x128 píxeles. Por otra parte, en la segunda técnica el volumen de densidad se crea de una manera explícita de abajo hacia arriba mediante votaciones probabilisticas emitidas por las características locales. En este caso nos encontramos los detectores basados en características.

  Por lo general, los inconvenientes de la búsqueda exhaustiva son el alto número de regiones candidatas a generar, lo que complica este tipo de métodos en aplicaciones que operan en tiempo real debido al alto coste computacional ya que el modelo de persona debe ser capaz de clasificar un gran número de negativos. A su favor cabe destacar la fortaleza de este tipo de algoritmos ante los cambios de escala o variaciones en la pose, lo que los hace muy aptos para escenarios complejos.

- **Segmentación y búsqueda exhaustiva:**

  Otra propuesta es combinar ambas técnicas para tratar de sumar sus puntos fuertes y resolver sus inconvenientes. Se realiza una selección de candidatos previos mediante segmentación y una segunda selección mediante búsqueda exhaustiva se aplica únicamente sobre los candidatos seleccionados. (cita algoritmos?)

2.1.2.2. Modelo de persona

Como se explicó anteriormente, el modelo de persona previamente entrenado se aplica en el proceso de clasificación o verificación sobre los candidatos resultantes del proceso de detección que, basándonos en las similitudes, se discriminarán como persona o no. Hay dos fuentes de información para caracterizar el modelo de persona: apariencia y movimiento.

- **Basados en movimiento:**

  Actualmente, la mayoría de los algoritmos están basados en información de apariencia o consiguen ser robustos frente al movimiento gracias al uso de algoritmos de seguimiento. Sin embargo, la apariencia es un factor sujeto a condiciones de iluminación, poses, vestimenta o el peso y altura de la persona. Por estas razones algunos enfoques van dirigidos en esta dirección con el fin de emplear únicamente información de movimiento.

  Mediante este método, [8] segmenta el movimiento y realiza un seguimiento de los objetos del frente. Los objetos son alineados a lo largo del tiempo y se calcula la similitud entre los objetos y como evoluciona en el tiempo. La propuesta de [9] se basa en detectar patrones de movimiento de las personas.

  En definitiva, en cuanto a la detección de personas, los algoritmos basados en movimiento obtienen peores resultados que los que se basan en información de apariencia. Por ello, pueden solamente ser utilizados como información complementario en escenarios específicos.

- **Basados en apariencia:**

  La información de apariencia es mucho más discriminatoria que la información de movimiento. Existen modelos simples de personas basados en modelos holísticos (referencias?). Por otro lado, modelos más complejos definen a las personas como combinación de varias regiones o formas, por ejemplo los modelos basados en partes [10]. Los más populares son aquellos que definen la apariencia de acuerdo con la información que se extrae de las características de bordes usando

algún descriptor de forma: HOG (*Histogramas de Gradientes Orientados*) [7] o ACF [1], que combina múltiples características.

- **Basados en apariencia y movimiento:**

  Aunque la mayoría de los algoritmos están basados en apariencia, algunos autores combinan apariencia y movimiento para mejorar los resultados en la detección. En [11] se combinan dos modelos de persona independientes: uno de ellos basado en apariencia y el otro basado en movimiento.

2.1.3. Fast Features Pyramids

El detector [1] empleado para el desarrollo de este trabajo está basado en el cálculo de lo que se conoce como Fast Features Pyramids. En esta sección el objetivo es profundizar en que consiste esta técnica y como podemos aprovecharla en la detección de personas.

2.1.3.1. Introducción

La descomposición en múltiples resoluciones y orientaciones es una de las técnicas fundacionales en el análisis de imágenes. Ha quedado demostrado que tales representaciones son las mejores para la extracción de información visual cuando se encuentran sobremuestreadas, y como prueba los estudios empíricos en visión por ordenador [7, 12]. Estas representaciones son muy robustas respecto a cambios de punto de vista, iluminación y deformaciones de la imagen.

El progreso en la detección de objetos durante la última década ha sido impresionante, consiguiendo que las tasas de falsos positivos hayan disminuido dos órdenes de magnitud. Desafortunadamente, esto ha venido acompañado de mayores costes computacionales lo que supone un factor muy relevante en aplicaciones que requieran tasas de detección en tiempo real. Fast Features Pyramids demuestra como es posible no pagar un gran precio computacional, la clave es aprovecharse de las características fractales de las imágenes para predecir fiablemente la estructura de la imagen a través de las escalas. Se estiman económicamente las características en un denso conjunto de escalas mediante la extrapolación de cálculos llevados a cabo en un conjunto de muestras de escalas. Esta visión conduce a tiempos de ejecución considerablemente reducidos, ACF consigue operar a más de 30 fps alcanzando unos grandes resultados en la detección de personas.

2.1.3.2. Histogramas de gradientes multiescala

Debido a que la estructura de una imagen se conserva cuando se sobremuestrea o submuestrea, se esperaría que sea posible aproximar los gradientes a una escala cercana a partir de los cálculos de gradientes de otra. Si esta afirmación se cumple podemos evitar el coste computacional de calcular los gradientes de cada escala en una pirámide de imágenes. El histograma de gradientes mide la distribución de los ángulos del gradiente en una imagen y se define por la magnitud del gradiente y su orientación:

$$M\,(i,j)^2 = \frac{\delta I}{\delta x}(i,j)^2 + \frac{\delta I}{\delta y}(i,j)^2$$

$$O(i,j) = \arctan\left(\frac{\delta I}{\delta y}(i,j)/\frac{\delta I}{\delta x}(i,j)\right)$$

Dada una imagen I y su imagen I' correspondiente al sobremuestreo o submuestreo por un factor dos y la magnitud del gradiente de las imágenes M y M', vemos en la imagen 2.2 cual es el comportamiento del ratio de sus gradientes.

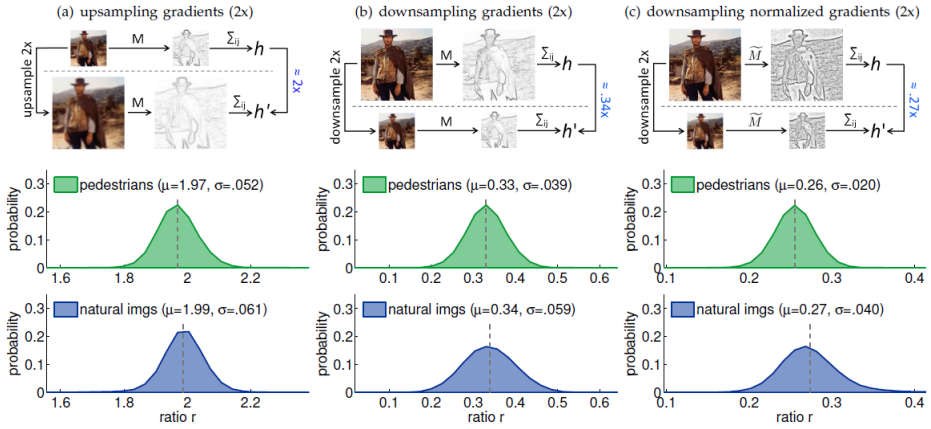

Figura 2.2: Comportamiento de los histogramas de gradientes en imagenes remuestreadas por un factor dos.

1. **Histograma de gradientes en imágenes sobremuestreadas**: La información contenida en una imagen sobremuestreada es similar a la que contenía la imagen original pero de menor resolución ya que el sobremuestreo no crea nuevas estructuras. Por tanto, la suma de las magnitudes de los gradientes de la imagen original y sobremuestreada deberían están relacionados por un factor de alrededor del factor de sobremuestreo. La orientación debería preservarse

también, esto nos permitirá aproximar histogramas de gradientes en imágenes sobremuestreadas usando los gradientes calculados en la escala original. Los experimentos se han realizado sobre un dataset de imágenes de escenas naturales y otro dataset de personas. Se ha definido como ratio la relación entre el histograma de gradientes de la imagen sobremuestreada y el histograma de gradientes de la imagen original. El resultado para ambos dataset arroja un ratio de aproximadamente 2 con una varianza relativamente pequeña y aunque gradientes individuales puedan cambiar, los histogramas estarán relacionados por la el valor del factor de muestreo.

2. **Histograma de grandientes en imágenes submuestreadas:** Mientras que la información en imágenes sobremuestreadas era similar a la información de la imagen original, para el caso de las imágenes submuestreadas nos encontramos con pérdida parcial de esta información original. Si la imagen no contiene apenas altas frecuencias la aproximación derivada del apartado anterior podrá ser utilizada. Sin embargo, el submuestreo significará pérdida de la energía de alta frecuencia de las imágenes que contengan y esto significará una diferencia de los gradientes. Si conseguimos una medida consistente para el ratio entre los histogramas podremos realizar una aproximación razonable. Los experimentos realizados dan como resultado un valor de 0.33 para el dataset de personas para imágenes submuestreadas por un factor 2.

3. **Histograma de gradientes normalizados:** Vamos a comprobar ahora como se comportan los histogramas de gradientes para un factor de submuestreo de 2 si utilizamos los gradientes normalizados. Los resultados reflejan un relación de 0.26 para el dataset de personas, lo que es un diferencia bastante significativa respecto al histograma original.

En la figura 2.3vemos cuales son los resultados al aproximar los gradientes por los valores obtenidos de los experimentos realizados. Los histogramas aproximados para las cuatro primeras imágenes son bastante buenos ya que apenas muestran variación respecto al histograma original. Las dos últimas imágenes han sido seleccionadas por contener estadísticas atípicas y mucha información de alta frecuencia y vemos como la aproximación para las imágenes submuestreadas falla.

2.1.3.3. Características multiescala

El siguiente análisis proporciona una comprensión del comportamiento de las características multiescala. Sea Ω cualquier función invariante de desplazamiento de

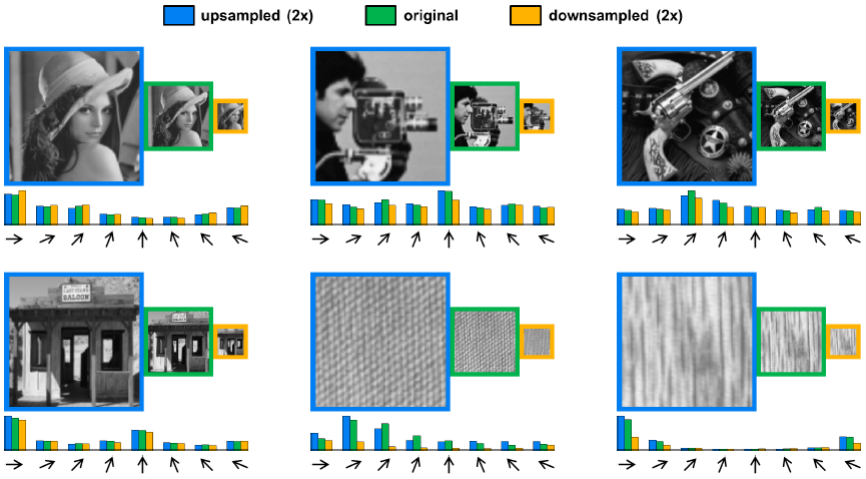

Figura 2.3: Histogramas de gradientes aproximados en imágenes remuestreadas por un factor dos.[1]

bajo nivel que toma una imagen I y crea una nueva imagen de canal $C = \Omega(I)$, donde C es un mapa de características por píxel tal que los píxeles de salida se calculan a partir de los píxeles de entrada en I. Definimos una característica $f_\Omega(I)$como la suma ponderada del canal C. Sea I_S la imagen I en una escala s, se define $f_\Omega(I_S)$ como la media global de C_S. Nuestro objetivo es entender como se comportan las características en función de la escala s.

En su trabajo [13], Ruderman y Bialek descubren que la relación entre las estadísticas de una imagen calculadas a dos escalas depende únicamente de la relación entre las escalas y que cada canal tendrá su correspondiente λ_Ω que podrá determinarse empíricamente.

$$f_\Omega(I_{S1}) / f_\Omega(I_{S2}) = (s_1/s_2)^{-\lambda_\Omega} + \varepsilon$$

2.1.3.4. Fast Feature Pyramids

El enfoque propuesto para construir la pirámide rápida de características se basa en las premisas introducidas anteriormente. En lugar de tener que calcular la imagen remuestreada para obtener el canal de características, se propone utilizar el canal de características de la imagen original para estimar el canal de la imagen remuestreada:

$$C_S \approx R(C, s) \cdot s^{-\lambda_\Omega}$$

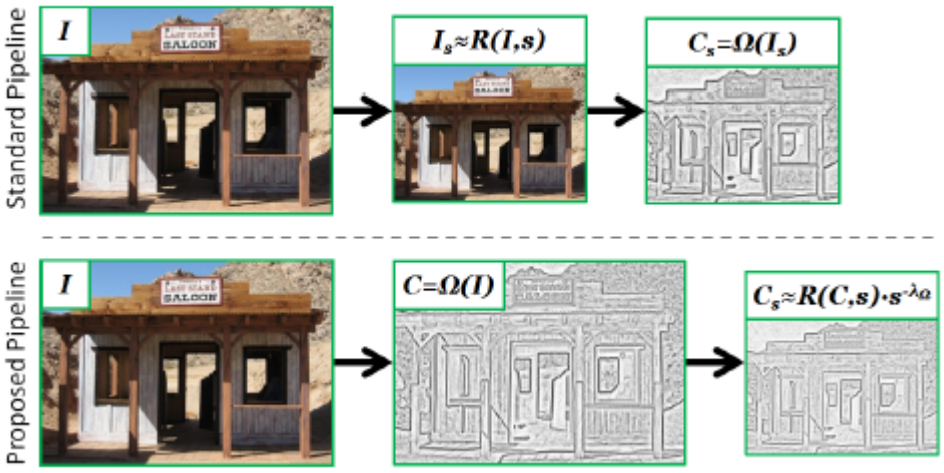

Figura 2.4: Cálculo del canal de características escalado.[1]

La pirámide de características es una representación multiescala de una imagen I donde los canales $C_S$son calculados para cada escala s, típicamente de 4 a 12 escalas por octava.

En la figura 2.5 podemos ver cual es el modelo propuesto para obtener los canales de características en las diferentes escalas. Para evitar grandes costes computacionales se empleará la aproximación, calculando solamente una imagen remuestreada por octava.

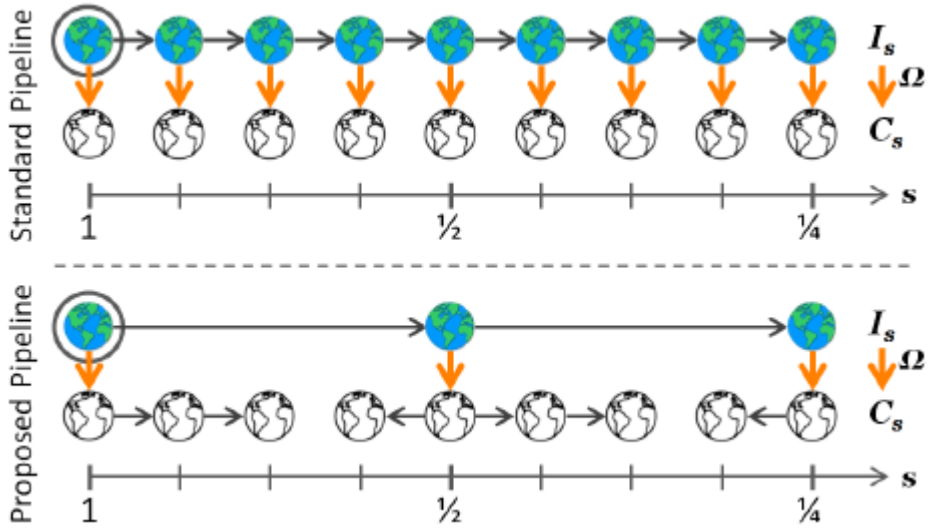

Figura 2.5: Construcción de la pirámide rápida de características. [1]

2.1.3.5. Aggregated Channel Features (ACF)

Dada una imagen de entrada I, calculamos los canales $C = \Omega(I)$, se realiza la suma de cada bloque de píxeles de C, suavizando los resultantes canales de baja resolución. Las características son búsquedas de un solo píxel en los canales agregados. Se utilizan los árboles de decisión sobre estas características para distinguir objetos del fondo empleando ventanas deslizantes multiescala.

Los canales que usa ACF son: magnitud de gradientes normalizados, histogramas de gradientes orientados (6 canales) y canales de color LUV. Antes de calcular los canales, la imagen I es suavizada usando un filtro [1 2 1]/4. Los canales se dividen en bloques de 4 x 4 y se suman los píxeles de cada bloque. Los canales resultantes son suavizados de nuevo por el mismo filtro. En detección de personas se utiliza AdaBoost [14] para el entrenamiento y conjugar 2048 arboles de decisión sobre las 5120 características candidatas en cada ventana de 128x64.

En la siguiente figura se resumen los resultados de la tasa de MR (log-average Miss Rate) en cuatro datasets.

	INRIA	Caltech	TUD	ETH	MEAN
Shapelet	82	91	95	91	90
VJ	72	95	95	90	88
PoseInv	80	86	88	92	87
HikSvm	43	73	83	72	68
HOG	46	68	78	64	64
HogLbp	39	68	82	55	61
MF	36	68	73	60	59
PLS	40	62	71	55	57
MF+CSS	25	61	60	61	52
MF+Motion	–	51	55	60	–
LatSvmV2	20	63	70	51	51
FPDW	21	57	63	60	50
ChnFtrs	22	56	60	57	49
Crosstalk	19	54	58	52	46
VeryFast	**16**	–	–	55	–
MultiResC	–	48	–	–	–
ICF-Exact	18	48	53	**50**	42
ICF	19	51	55	56	45
ACF-Exact	17	**43**	**50**	**50**	**40**
ACF	17	45	52	51	41

Figura 2.6: MR para detección de personas en cuatro datasets. [1]

Capítulo 3

Diseño y desarrollo

3.1. Introducción

En este capítulo vamos a explicar en detalle como funciona el algoritmo base [1], a partir del cual hemos incorporado las modificaciones para mejorar el rendimiento de detección de personas, y el algoritmo de adaptación propuesto. Este último ha sido diseñado para se capaz de extraer información de la escena, de tal forma que la adaptación a los diferentes escenarios nos lleve a obtener mejores resultados de detección.

3.2. Algoritmo base

3.2.1. Diseño

El detector utilizado implementa el código de detección Aggregate Channel Features (ACF) que vimos anteriormente en 2.1.3.5. El detector ACF es un rápido y efectivo detector de ventana deslizante, trabaja a 30 fps, y es la evolución del detector Viola & Jones (VJ) [15] pero con una disminución de falsos positivos de tres ordenes de magnitud.

En la figura 3.1 podemos ver cual es el esquema del detector. En primer lugar recibe una imagen o frame y se calculan los correspondientes canales, se suma cada bloque de píxeles del canal y realiza un filtrado para suavizar los canales resultantes de menor resolución. Las características son búsquedas de un solo píxel en los canales agregados. Se utilizan los árboles de decisión sobre estas características para distinguir objetos del fondo empleando ventanas deslizantes multiescala.

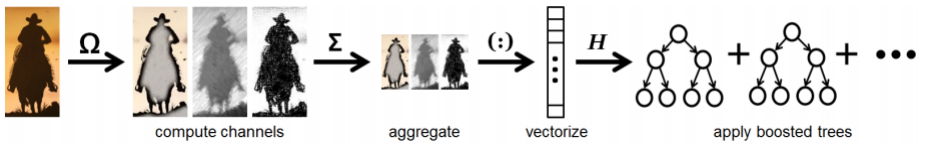

Figura 3.1: Visión conjunta del esquema de funcionamiento del detector ACF

3.2.2. Desarrollo

Veremos ahora un análisis mas detallado de las diferentes partes que componen código del detector base utilizado, centrándonos especialmente en el cálculo rápido de los canales de características descritos en [16, 17, 18, 1].

3.2.2.1. Canales

AFC hace uso 10 canales: magnitud de gradiente normalizado, histograma de gradientes orientados (6 canales) y canales de color LUV. Pero antes de calcular los canales de la imagen se aplicará a esta un filtro de suavizado [1 2 1]/4.

Para calcular los canales en una escala utilizamos la función *chns = chnsCompute(I, varargin)*. Un ejemplo de canal simple sería la imagen en escala de grises o cada canal de color de una imagen a color. Los tres canales que más efectivos han resultado en la detección mediante ventana deslizante son: canales de color, magnitud de gradiente y los canales de histograma de gradientes.

- Canales de color: La función *J = rgbConvert(I, colorSpace, useSingle)* será la encargada de convertir una imagen RGB de entrada a otro espacio de color. Por defecto el espacio de color que emplea la función es: *colorSpace=='luv'*. Este espacio es perceptualmente uniforme y los canales LUV se corresponden respectivamente con la imagen de luminancia, verde-rojo, y azul-amarillo. Estos canales se normalizan para que su valor esté dentro del rango [0,1].

- Canal de magnitud del gradiente: *[M,O] = gradientMag(I, channel, normRad, normConst, full)* calcula la magnitud y orientación del gradiente en cada posición de la imagen. Esta función puede hacer uso del canal de color obtenido anteriormente para el cálculo del gradiente, aunque por defecto estará deshabilitado.

- Canales de histograma de gradientes: *H = gradientHist(M, O, varargin)*. La idea de esta técnica es que la apariencia de un objeto se puede caracterizar bastante bien por la distribución de los grandientes de intensidad locales o

por la dirección de sus bordes. La entrada de la función son la magnitud y orientación del gradiente que previamente ha devuelto la función anterior. Por cada región de la imagen de entrada se calcula un histograma de gradientes, con cada gradiente cuantificado por su orientación y ponderado por su magnitud. El primer paso es la cuantificación de la magnitud M en los canales de orientación de acuerdo con su orientación O. La magnitud de cada localización se localiza las dos orientaciones más cercanas usando interpolación lineal. El resultado es un mapa de características que representa los histogramas de gradiente en cada región de la imagen 3.2. Estas características generalizan las características HOG introducidas en [7].

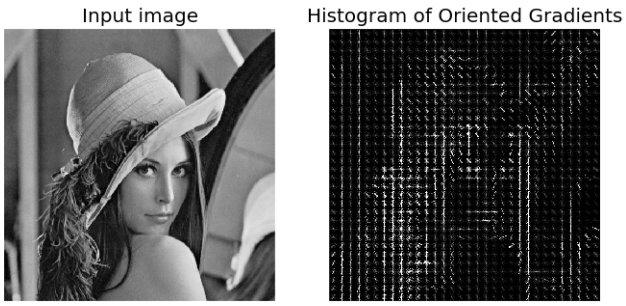

Figura 3.2: Mapa de características de los histogramas de gradientes

3.2.2.2. Pirámide de canales de características

Mientras que la función descrita en 3.2.2.1 calculaba los canales de características en una escala, la función *pyramid = chnsPyramid(I, varargin)* llama múltiples veces a *chnsCompute()* en diferentes escalas de la imagen para crear una pirámide espacio-escala de los canales de características.

Primeramente se crea una pirámide de imagen y entonces se hace la llamada a la función *chnsCompute()* con los parámetros de entrada correspondientes para cada escala de la pirámide de imágenes. El número de escalas por octava lo determina el parámetro *nPerOct*, que será típicamente 8. La menor y mayor escala también vendrá determinado en la entrada, la creación de la pirámide se detendrá cuando la dimensión de la imagen sea menor mientras que la escala mayor determinará el número de octavas que se calculan sobre la imagen original.

La idea de construir las pirámides rápidas de características (*Fast Features Pyramid*) supone que sea necesario calcular los canales en cada escala de la imagen. Como

ya vimos anteriormente en 2.1.3.4, las características de una escala pueden ser utilizadas para aproximar la respuesta de las características en las escalas próximas. El parámetro *nApprox* determina cuantas escalas intermedias son aproximadas utilizando la técnica descrita, por ejemplo si el número de escalas por octava es *nPerOct=8* y *nApprox=7* entonces las 7 escalas intermedias serán aproximadas, remuestreada y normalizadas adecuadamente. El parámetro *lambda* será el que nos indique como debe ser la normalización de los canales. Normalmente aproximar todas las escalas en una octava funciona bastante bien y da como resultado una velocidad de 4 veces mayor sobre el cálculo de la pirámide sin aproximaciones.

3.2.2.3. Detector

Con todo esto, veremos ahora como funciona el algoritmo base de detección. En primer lugar extraen las imágenes de entrenamiento y test del dataset empleado para el trabajo: INRIA Person Dataset [7], así como las anotaciones (ground truth). Para el entrenamiento, el dataset contiene imágenes negativas sin personas en las escenas y un directorio de imágenes positivas asociadas a sus correspondientes anotaciones de donde se sitúa la persona. Como resultado del entrenamiento se genera un modelo que será utilizado en detección para la localización de personas en los frames de la secuencia de vídeo. En la figura 3.3vemos un ejemplo del resultado de la detección sobre una imagen. Cada blob de detección que se genera sobre la imagen tiene una puntuación que determina con cuanta fiabilidad se tratará de una persona. Con los resultados de la detección sobre las diferentes imágenes y haciendo uso de las anotaciones proporcionadas se realiza la fase de test del detector, el test comprobará que que se hayan detectado las personas presentes en la imágenes y cual es la tasa de FPPI (false positives per image) registrada (detecciones de personas sobre una región sin persona). La imagen 3.4 nos muestra cual es el resultado de la fase de test.

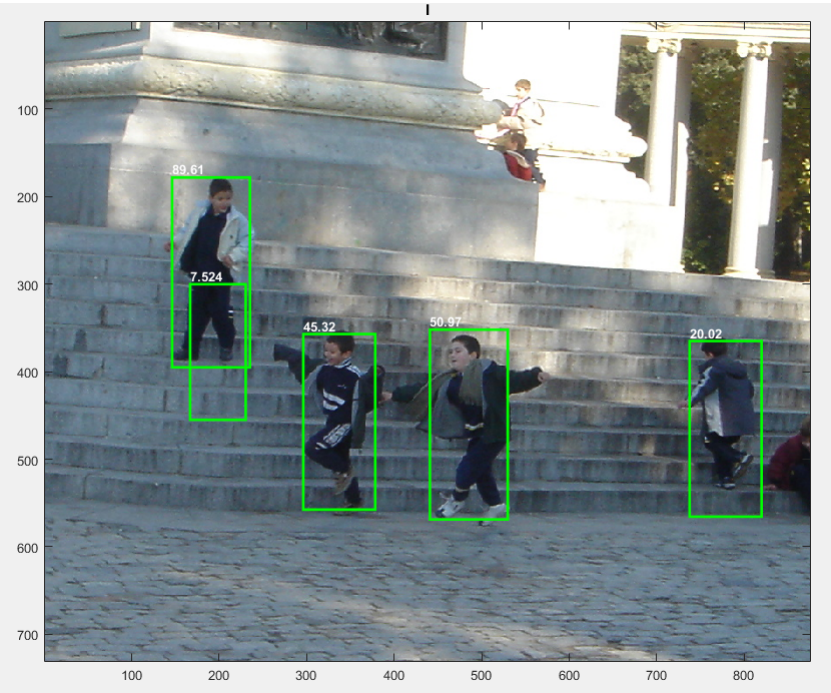

Figura 3.3: Resultado de la detección sobre un frame de ejemplo utilizando el detector base.

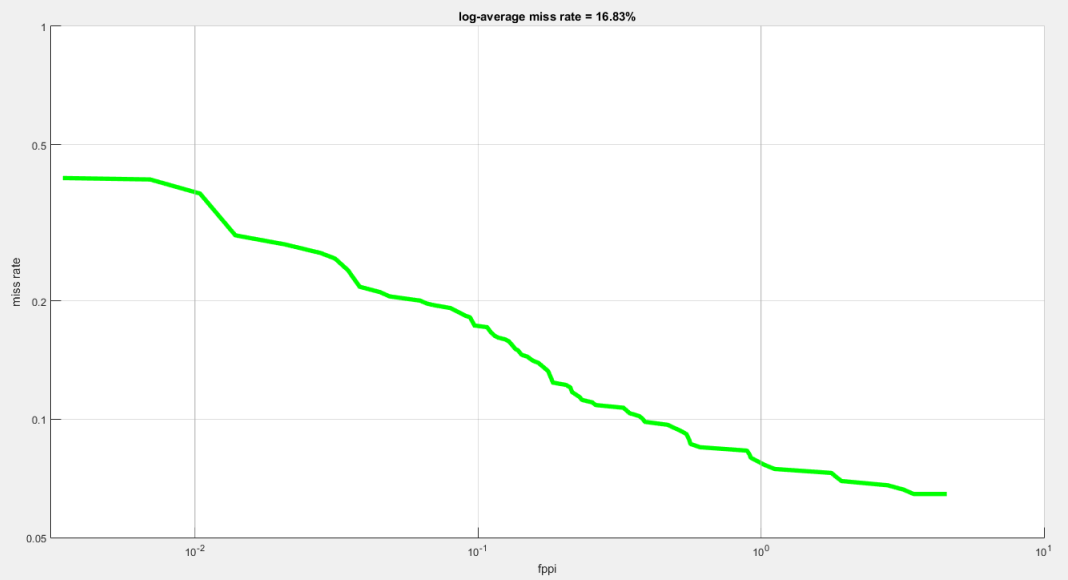

Figura 3.4: Tasa de error y falsos positivos por imagen del detector base.

3.3. Algoritmo propuesto

3.3.1. Adaptación del algoritmo a la escena

Recordemos que como objetivo del trabajo se propone conseguir que el algoritmo de detección consiga adaptarse a la multitud de escenarios variados que pueden darse en diferentes situaciones y lugares. Para satisfacer esto será necesario que en la detección podamos incorporar información sobre el contexto de la escena. Si sabemos aprovechar el conocimiento que tenemos del escenario conseguiremos obtener unos resultados de detección mucho mejores incluso en las secuencias de vídeo más complejas.

En primer lugar y para demostrar que esta idea va bien orientada, proponemos una versión *Dummy* para adaptar el algoritmo. Una vez que comprobamos que vamos por buen camino, la versión *Dummy* dejará paso a una versión mas realista de la propuesta que genera un modelo con la propia escena como fuente de información.

3.3.2. Diseño Dummy

Esta primera versión se trata de una aproximación a la idea para adaptar el algoritmo a los escenarios. La fase de entrenamiento del detector es crítica, por lo que disponer de un buen modelo de persona nos asegurará obtener resultados en la detección fieles a la realidad. Este modelo de persona se calcula a partir de los ejemplos positivos y negativos de entrenamiento que proporciona el dataset de INRIA Person. El dataset cuenta con un gran número de imágenes de entrenamiento en diferentes situaciones pero, evidentemente, no contendrá los mejores ejemplos para escenas concretas. Esto supone que en escenarios complejos disminuyan drásticamente los resultados de detección, ya que poses complicadas, oclusiones o puntos de vista poco comunes no estarán completamente representados en el dataset.

Un buen primer acercamiento propuesto para demostrar la idea sería generar en la fase de entrenamiento un modelo que se genere contando con las anotaciones (ground truth) de las secuencias de vídeo a analizar. Si conocemos las posiciones en cada frame donde se encuentran las personas, y por tanto también las regiones en las que no aparecen, el modelo construido durante el entrenamiento será extremadamente preciso y adecuado a la escena.

3.3.2.1. Desarrollo

Primeramente preparamos un dataset auxiliar que contendrá la información al completo del dataset de INRIA Person más las imágenes negativas y positivas que

se deducen de las anotaciones del vídeo. Tenemos por tanto tres vías que exploraremos para conocer con cual de ellas obtener mejores resultados: Incorporar únicamente imágenes positivas para el entrenamiento, esto es obtener los frames en los que existe una anotación de las personas, añadir la imagen como ejemplo positivos del dataset con sus correspondientes anotaciones; Incorporar únicamente imágenes negativas para el entrenamiento, es decir, descartar las frames en los que aparecen anotadas regiones como personas y quedarnos con el resto de cuadros sin anotaciones como ejemplos negativos: Incorporar al dataset tanto los ejemplos positivos como los negativos. Se muestra la imagen 3.5 como ejemplos positivos y negativos que se incorporan al dataset para un escenario concreto:

Figura 3.5: Imágenes incorporadas al dataset de entrenamiento como ejemplos positivo (izq.) y negativo (dcha.).

A pesar de ofrecer un resultados mucho mejores que el algoritmo base, no se trata de una implementación realista ya que estamos contando de partida con las anotaciones que indican la localización de las personas en cada frame de la secuencia. Sin embargo, los resultados nos abren una puerta a seguir investigando por esta vía y así será como llegaremos al siguiente método propuesto para cumplir con la finalidad del trabajo.

3.3.3. Caltech Pedestrian Dataset para generar dataset de entrenamiento

En esta sección se profundiza en la idea desarrollada en 3.3.2 desde una perspectiva más realista y realizable. El problema con el que nos encontrábamos en el diseño *Dummy* es que es necesario disponer de las anotaciones de los vídeos sobre los que vamos a detectar. Por regla general tiene sentido que esto no sea así ya que el objetivo de la detección será encontrar la localización de las personas sin conocer que estén

presentes en la imagen. Esto nos conduce a la necesidad de encontrar la manera de generar los ejemplos positivos y negativos de la imagen para incorporarlos al dataset empleado en el entrenamiento.

Lo que proponemos en este trabajo es construir un modelo a partir de un dataset diferente que nos permita hacer una detección previa de los candidatos a persona. Los resultados de esta detección nos darán la información necesaria para determinar las regiones que muy probablemente sean personas o no. Una vez que contamos con los datos de la detección y a partir del dataset INRIA Person, generamos un nuevo dataset sobre el cual entrenar y evaluar la detección en el vídeo.

3.3.3.1. Desarrollo

Debemos comenzar en este caso por construir el modelo desde los datos de entrenamiento de un datset distinto al del entrenamiento inicial, en este caso el dataset Caltech Pedestrian [19]. Este modelo calculado nos permitirá realizar la detección previa para incorporar las imágenes de positivos y negativos al modelo del punto de partida a actualizar. Será necesario tener especial cuidado con las imágenes que se agregan como datos específicos del escenario ya que incluir falsos positivos será contraproducente.

Para hacer una buena selección de imágenes de entrenamiento debemos tener en cuenta cuales son las calificaciones que reciben los blobs. Las notas más elevadas serán casi con total seguridad imágenes positivas mientras que las más bajas se corresponderán con regiones sin presencia de personas. Para realizar esta selección se ha desarrollado la función *[Blobs, Scores, minpos, minneg] = PercentileCaltech(video_dir, video_names, percentile)*. Recibe de entrada el directorio donde se localizan los vídeos, la secuencia sobre la que estamos trabajando y el valor del percentil que nos permita quedarnos con las muestras más válidas para el entrenamiento. El funcionamiento es sencillo y conseguiremos asegurar que la selección que realicemos de positivos y negativos sea fiable. La función *ReadACFBlobs* utilizando el modelo calculado con el dataset de Caltech nos devolverá un cell-array con los blobs detectados y un vector de las puntuaciones. Construimos un histograma de las puntuaciones obtenidas, el valor de *percentile* nos indicará con que porcentaje de las puntuaciones quedarnos. Con un *percentile* de 5 seleccionaremos las calificaciones contenidas en el 5 % mayor y menos del histograma para los ejemplos positivos y negativos respectivamente. Gracias a esto contamos con cual será la nota mínima que deben cumplir los blobs para ser incorporados al dataset como positivos y la nota máxima hasta la cual tomaremos los blobs como negativos.

La función *AddInDatabasePos* recibirá como parámetros de entrada el cell-array

Blobs que contiene las coordenadas en la imagen, el número de frame y la puntuación. En cada frame que existan blobs comprobaremos si la puntuación de estos es mayor o igual que el valor mínimo que nos asegurará que sea una persona. Si esta condición se cumple el frame se incorpora al dataset como imagen positiva y se escribe su correspondiente anotación que localiza a la persona.

Mediante *AddInDatabaseNeg* completamos el dataset con los ejemplos negativos para cada frame de la secuencia. Por cada imagen podremos tener uno, varios o ningún blob. Aquellos cuadros para los que no exista blob pasarán directamente como negativos al dataset de entrenamiento, mientras que si hay algún blob presente habrá que comprobar si su puntuación es mayor o menor al límite establecido. Existe un valor hasta el cual consideraremos poco probable que ese blob sea una persona por su baja puntuación, y por encima de este límite no podremos estar tan seguros. Es por esto que todos aquellos blobs que tengan una nota superior al límite establecido se eliminarán de la imagen, el valor de sus píxeles pasará a ser cero y de esta forma nos quedará una imagen que sólo contendrá regiones negativas. La imagen 3.6 muestra un frame de ejemplo como imagen negativa incorporada al dataset para unos percentiles de 5 y 20.

Figura 3.6: Imágenes negativas a partir de un mismo frame con percentil 5 (dcha.) y 20 (izq.).

Vemos como para un percentil menor la confianza es también menor y se eliminan de la imagen regiones con puntuaciones en un amplio rango, muchas de ellas no serán persona y contendrán información del fondo de la escena. Esto tiene el inconveniente de no incluir parte del contexto de la escena que no sería útil para adaptar el algoritmo, pero por otra parte estaremos seguros de que los ejemplos negativos y positivos agregados para generar el modelo serán fiables.

Una vez hemos incorporado al dataset INRIA los nuevos ejemplos positivos y

negativos, se calcula el modelo que empleará el detector. Este modelo generado contendrá información característica de cada escena, los ejemplos positivos nos servirán para incluir la apariencia de las personas y punto de vista de la cámara. Por otra parte, los ejemplos negativos incluirán partes propias de la escena como fondo u objetos móviles no caracterizados como personas que serán determinantes para no clasificar estas partes como personas.

Capítulo 4

Evaluación

4.1. Introducción

Una vez desarrollado el algoritmo propuesto, en este capítulo vamos a evaluar los resultados obtenidos. En primer lugar se describe el marco de evaluación, donde se detallarán los dataset utilizados y las métricas de evaluación propuestas en [4]. A continuación se expondrán los resultados obtenidos y se analizarán comparándolos con los resultados del algoritmo base.

4.2. Marco de evaluación

4.2.1. *Dataset*

Dentro del proceso de evaluación, el primer paso es disponer de una base de datos sobre la que evaluar los resultados. En nuestro caso los dos dataset empleados han sido INRIA Person Dataset [7] y Caltech Pedestrian Dataset [19].

- INRIA Person Dataset: Realizamos la fase de entrenamiento del algoritmo base y la fase de test de nuestro detector utilizando el dataset INRIA, el cual está formado por 1805 imágenes de 64x128. Estas imágenes han sido recortadas de un conjunto variado de fotografías personales del autor. Las personas normalmente aparecen de pie y pueden encontrarse en cualquier orientación, además de contar con una amplia variedad de imágenes de fondo. Las carpetas 'Train' y 'Test' contienen respectivamente imágenes de entrenamiento y test. Ambas carpetas tienen tres subdirectorios: 'pos' (imágenes positivas), 'neg' (imágenes negativas), y 'annotations' (archivo de anotaciones e las imágenes positivas).

- Caltech Pedestrian Dataset: La base de datos Caltech Pedestrian nos ha servido para apoyarnos en una fuente de información diferente a la hora de generar un modelo que aproveche el contexto de la escena. Dispone de 250,000 frames capturados desde una cámara en un vehículo en movimiento. Se han anotado un total de 350,000 blobs sobre unas 2,300 personas. Por cada frame en el que una persona es visible, se anota un blob que indique a la persona en su totalidad, en caso de existan oclusiones se anota un segundo blob sobre la región visible.

Las figuras 4.1y 4.2 muestran tres imágenes aleatorias de ejemplo de cada dataset.

Figura 4.1: Tres imágenes positivas de ejemplo del dataset INRIA

Figura 4.2: Tres imágenes de ejemplo del dataset Caltech

4.2.2. Métricas de evaluación

Para la evaluación del rendimiento de los algoritmos utilizaremos el sistema del laboratorio de investigación VPULab (*Video Processing and Understanding Lab*). La manera de probar el rendimiento de un detector es ejecutarlo sobre una secuencia, conociendo sus correspondientes anotaciones de Ground Truth, y comparar los resultados. En función de la hipótesis del detector y las anotaciones, los resultados podrán clasificarse como: TP (*True Positive*) si el sistema detecta persona y acierta; TN (*True Negative*) si el sistema no detecta persona y acierta; FP (*False Positive*) si el sistema detecta persona y se equivoca; FN (*False Negative*) si el sistema no detecta persona y se equivoca.

El rendimiento puede ser evaluado en dos niveles: a nivel de frame o a nivel de secuencia global. El rendimiento a nivel de frame o ventana suele medirse en términos de DET (Detection Error Tradeoff) [7, 19] o mediante la curva ROC (Receiver Ope-

rating Characteristics). El rendimiento de la secuencia global se mide según la curva Precision-Recall. El primer caso nos aporta información sobre la etapa de clasificación, mientras que el segundo proporciona información del rendimiento del sistema completo. Desde la perspectiva de los sistemas de videovigilancia es más interesante comparar sobre el rendimiento general, por lo que la métrica utilizada sera Precision-Recall.

El resultado del sistema de detección serán unas puntuaciones denominadas score que indicarán la confianza de que los objetos detectados sean realmente lo que se buscaba. A mayor score, mayor probabilidad de que el objeto detectado sea el deseado. El método de evaluación compara los parámetros precisión y recall para diferentes umbrales y cada umbral representa un punto de la curva.

Precisión y Recall se calcula según las ecuaciones siguientes:

$$Precision = \frac{\#TruePositivePeopleDetections}{\#TruePositivePeopleDetections + \#FalsePositivePeopleDetections}$$

$$Recall = \frac{\#TruePositivePeopleDetections}{\#TruePositivePeopleDetections + \#FalseNegativePeopleDetections}$$

Precisión es la relación entre el número total de detecciones reales (TP) y el número de detecciones (TP+FP). Recall es la relación entre las deteciones reales (TP) y el número de objetos que deberían haber sido detectados idealmente. El rendimiento del sistema será mejor cuanto más se acerquen ambos valores a uno. También se medirá el área bajo la curva que permite evaluar los parámetros de precisión y recall, a mayor área mejor rendimiento.

4.3. Pruebas y resultados

4.3.1. Pruebas y resultados del algoritmo base frente a la versión Dummy

El objetivo de esta evaluación es comprobar que la idea que tomamos de partida es buena y mejora el rendimiento del detector. Las pruebas realizadas han sido:

1. DummyPos: Adaptar el algoritmo incluyendo como información de la escena las imágenes positivas con sus correspondientes anotaciones de la secuencia.

2. DummyNeg: Adaptar el algoritmo incluyendo como información de la escena las imágenes negativas de la secuencia.

3. Dummy: Adaptar el algoritmo incluyendo como información las imágenes positivas y negativas de la secuencia.

Las pruebas se han realizado sobre un conjunto de 19 secuencias [20]. Se muestra como ejemplo la representación Precisión-Recall sobre la secuencia *abandonedBox* en la figura 4.3. El resto de figuras se presentan en A.

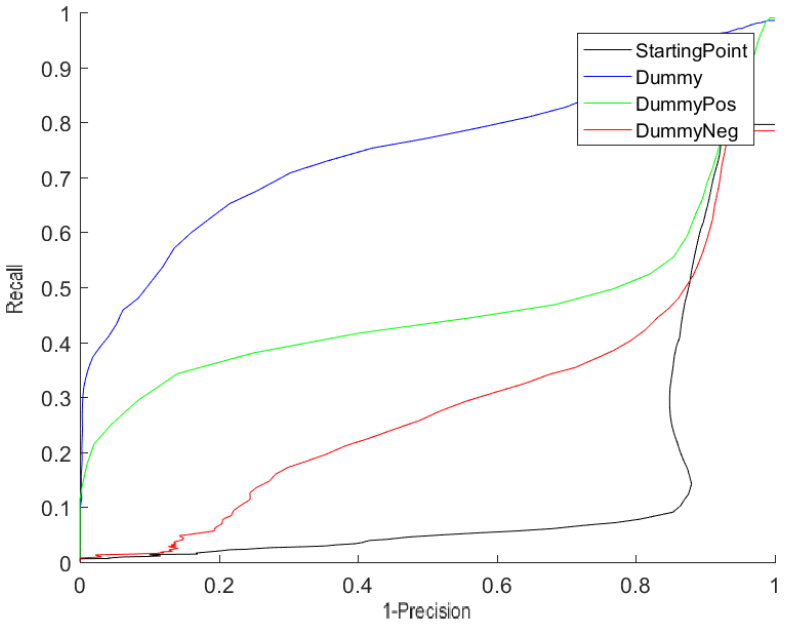

Figura 4.3: Curva Precisión-Recall sobre la secuencia 'abandonedBox' utilizando la versión Dummy

Vemos como de partida, el rendimiento del algoritmo base *StartingPoint* sobre la secuencia es bastante malo. Si observamos la secuencia 4.4 esto puede deberse a las oclusiones que permanentemente hay en el vídeo, a las sombras en una parte de la imagen y al punto de vista desde el que está grabado. Si incluimos la información sobre la escena a modo de ejemplos positivos y negativos con las anotaciones de la secuencia como fuente de información el resultado es mucho mejor al evaluar el algoritmo adaptado.

La siguiente tabla resumen 4.1 muestra el rendimiento de los algoritmos para cada una de las secuencias y el porcentaje de mejora respecto al punto de partida.

Figura 4.4: Frame de ejemplo de la secuencia 'abandonedBox'

Evaluación Ground Truth (% área bajo la curva)				
Secuencia	StartingPoint	Dummy	DummyPos	DummyNeg
PETS2006	80.37	98.43	97.83	83.94
abandonedBox	13.29	74.97	46.14	28.24
backdoor	93.92	96.53	96.21	93.06
badminton	74.67	88.06	70.90	71.29
busStation	92.93	97.78	97.72	93.71
copyMachine	56.05	96.64	97.23	48.15
cubicle	81.71	78.71	72.32	81.87
fall	35.52	89.57	73.34	55.87
office	99.68	100	100	99.93
overpass	49.59	92.06	76.94	73.69
pedestrians	91.71	96.52	93.91	91.46
peopleInShade	94.80	99.10	97.93	97.83
sidewalk	35.59	59.32	53.49	30.84
skating	84.77	93.97	91.59	86.77
sofa	90.28	100	93.13	90.74
tramstop	20.41	86.38	59.77	20.51
wetSnow	7.82	91.16	54.78	11.32
winterDriveway	28.49	84.83	74.32	26.82
zoomInZoomOut	35.52	79.19	56.43	44.22
Media	**61.42**	**85.09**	**76.01**	**63.65**

Tabla 4.1: Evaluación Ground Truth del algoritmo Dummy

4.3.2. Pruebas y Resultados del algoritmo propuesto

A la vista de los resultados anteriores, buscar una solución más realista al problema continuando con la propuesta es una buena idea. El algoritmo propuesto se ha evaluado variando como parámetro de entrada el valor del percentil, que como vimos en 3.3.3.1 nos dará un mayor o menos grado de confianza para seleccionar las muestras para la adaptación.

Hemos realizado pruebas para percentiles desde 5 % hasta 15 %. El mejor valor vendrá determinado por las características propias de cada secuencia. La imagen 4.5 muestra el resultado de la evaluación sobre la secuencia *abandonedBox*, al igual que en la sección anterior, y el resto de pruebas pueden verse en A.

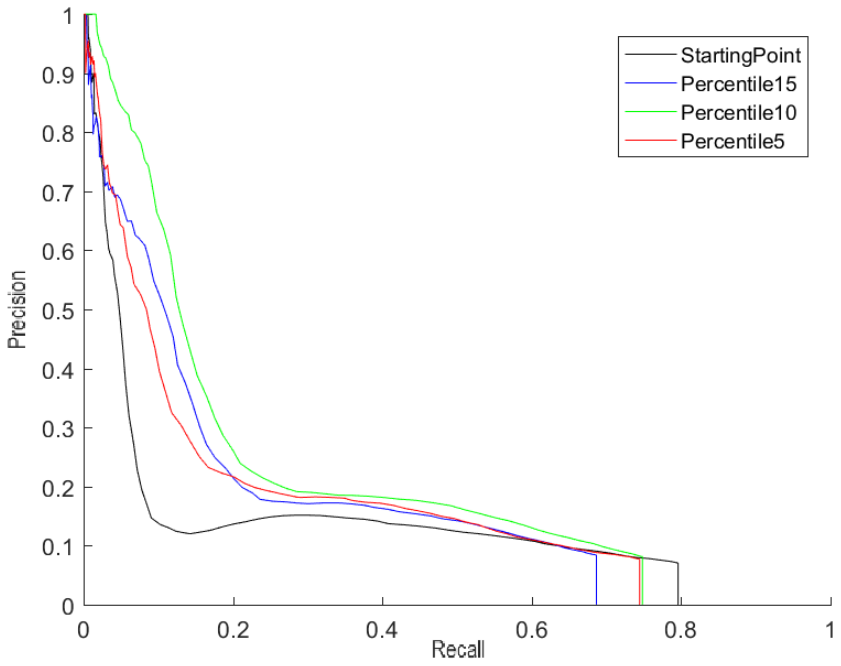

Figura 4.5: Curva Precisión-Recall sobre la secuencia 'abandonedBox' utilizando el algoritmo propuesto

La tabla 4.2 muestra el resumen del porcentaje del área bajo la curva en cada una de las secuencias para medir el rendimiento de nuestro algoritmo propuesto.

Evaluación Ground Truth (% área bajo la curva)				
Secuencia	StartingPoint	Percentil 5	Percentil 10	Percentil 15
PETS2006	80.37	77.66	79.63	78.6
abandonedBox	13.29	17.52	21.54	17.27
backdoor	93.92	93.76	94.02	92.81
badminton	74.67	73.96	72.62	72.63
busStation	92.93	93.70	93.13	92.98
copyMachine	56.05	54.56	53.10	54.95
cubicle	81.71	88.35	82.87	85.80
fall	35.52	66.89	41.68	28.30
office	99.68	100	100	100
overpass	49.59	64.38	61.81	54.19
pedestrians	91.71	92.22	93.21	92.23
peopleInShade	94.80	93.49	93.15	92.26
sidewalk	35.59	56.42	49.65	39.52
skating	84.77	84.21	84.80	85.25
sofa	90.28	90.46	90.95	89.49
tramstop	20.41	21.35	18.47	25.43
wetSnow	7.82	26.61	17.32	9.32
winterDriveway	28.49	67.76	60.20	15.12
zoomInZoomOut	35.52	39.70	33.84	41.58
Media	**61.42**	**68.47**	**65.37**	**61.45**

Tabla 4.2: Evaluación Ground Truth del algoritmo propuesto

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

El objetivo principal del trabajo era aprovechar la información de contexto de la escena para incorporarla al detector ACF [1] descrito en el estado del arte. Para conseguir el propósito se ha implementado un algoritmo que permite obtener de la escena frames positivos y negativos en base a los cuales adaptar el sistema y detectar personas utilizando esta información de contexto.

Para llegar a esto, primeramente se ha realizado un estudio general del estado del arte de la detección de personas y despues concretamente del algoritmo base a desarrollar. A continuación se creó un algoritmo sencillo que utilizase las anotaciones de Ground Truth como fuente de información con la idea de demostrar que la propuesta de partida podría ser buena. A partir del algoritmo base se desarrolla el algoritmo que proponemos en este trabajo, el cual es capaz de generar un nuevo modelo utilizando información extraída de los frames de la secuencia. Este modelo conseguirá adaptar el sistema de detección en cada escenario concreto.Por último se ha estudiado el criterio de evaluación para los detectores de personas y se ha evaluado el rendimiento del algoritmo propuesto.

A la vista de los resultados, podemos concluir que la mejora es muy significativa en aquellas secuencias para las que el detector presentaba un rendimiento malo, véase B.7, B.9, B.12, B.16, B.17. Suelen ser videos con bastantes oclusiones, donde existen personas que aparecen lejanas en la escena o demasiado cerca como para verse completamente, con puntos de vista poco frecuentes, etc. Para los casos en los que los resultados de partida eran ya muy buenos es complicado mejorar el rendimiento. En algunos casos se realizan ligeras mejoras, como en B.4, B.6, B.8, B.10, B.18, B.15, B.14 y B.13, pero debemos tener cuidado para no introducir nueva información que

no se ajuste exactamente, especialmente si son pocos los ejemplos que incorporamos (tantos positivos como negativos). Esto puede hacer disminuir el rendimiento levemente en B.1, B.2, B.3, B.11 y B.5. En media, los resultados obtenidos refuerzan nuestra hipotesis de partida, pudiendo demostrar la mejora en las detecciones que supone la adaptación utilizando el contexto de la escena.

5.2. Trabajo futuro

Partiendo del trabajo realizado, es evidente la necesidad de continuar trabajando en el desarrollo de algoritmos de detección de personas con el fin de mejorar su rendimiento en situaciones complejas.

El algoritmo propuesto tiene el inconveniente de que no podrá trabajar en tiempo real, algo de especial importancia en ciertos sectores, ya que es preciso contar de antemano con las imágenes para extraer de ellas la información y generar el modelo para el sistema adaptativo. Por eso una de la líneas que se propone como trabajo futuro es conseguir que la adaptación funcione en tiempo real o que apenas suponga retraso. Una opción podría ser utilizar únicamente un conjunto de primeros frames y calcular el modelo en base a esa información.

Otra vía de desarrollo puede ser tratar de estimar cual sería un percentil óptimo para cada secuencia, ya que puede variar en gran medida de esta decisión. De esta forma, conseguiremos optener para cada escenario concreto el mejor resultado posible aplicando la idea de este trabajo.

Proponemos también valorar el posible estudio de algoritmos de seguimiento o *tracking*, que son más robustos frente a las oclusiones y permiten seguir a múltiples objetivos. Con la incorporación de información de movimiento es más que probable que mejore el rendimiento y añada robustez a la detección.

Bibliografía

[1] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 1532–1545, Aug 2014.

[2] The Mathworks, Inc., Natick, Massachusetts, *MATLAB version 8.5.0.197613 (R2015a)*, 2015.

[3] L. G. Roberts, "Machine perception of three-dimensional solids," *Massachusetts Institute of Technology*, 1963.

[4] A. Garcia-Martin and J. M. Martinez, "People detection in surveillance: classification and evaluation," *IET Computer Vision*, vol. 9, no. 5, pp. 779–788, 2015.

[5] A. Garcia-Martin and J. M. Martinez, "Robust real time moving people detection in surveillance scenarios," in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 241–247, Aug 2010.

[6] N. Sprague and J. Luo, "Clothed people detection in still images," in *Object recognition supported by user interaction for service robots*, vol. 3, pp. 585–589 vol.3, 2002.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, June 2005.

[8] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 781–796, Aug 2000.

[9] H. Sidenbladh, "Detecting human motion with support vector machines," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, pp. 188–191 Vol.2, Aug 2004.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1627–1645, Sept 2010.

[11] l. Garcia-Martin and J. M. Martinez, "On collaborative people detection and tracking in complex scenarios," *Image and Vision Computing*, vol. 30, no. 4, pp. 345–354, 2012.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[13] D. L. Ruderman and W. Bialek, "Statistics of natural images: Scaling in the woods," in *Advances in Neural Information Processing Systems 6* (J. D. Cowan, G. Tesauro, and J. Alspector, eds.), pp. 551–558, Morgan-Kaufmann, 1994.

[14] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Statist.*, vol. 28, pp. 337–407, 04 2000.

[15] M. Viola, M. J. Jones, and P. Viola, "Fast multi-view face detection," in *Proc. of Computer Vision and Pattern Recognition*, 2003.

[16] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," 2009.

[17] P. Dollar, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *Proceedings of the British Machine Vision Conference*, pp. 68.1–68.11, BMVA Press, 2010. doi:10.5244/C.24.68.

[18] P. Dollar, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *Computer Vision–ECCV 2012*, pp. 645–659, Springer, 2012.

[19] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[20] A. Garcia-Martin, B. Alcedo, and J. M. Martinez, "Pdbm: people detection benchmark repository," *Electronics Letters*, vol. 51, no. 7, pp. 559–560, 2015.

Apéndice A

Resultados del algoritmo base frente a la versión Dummy

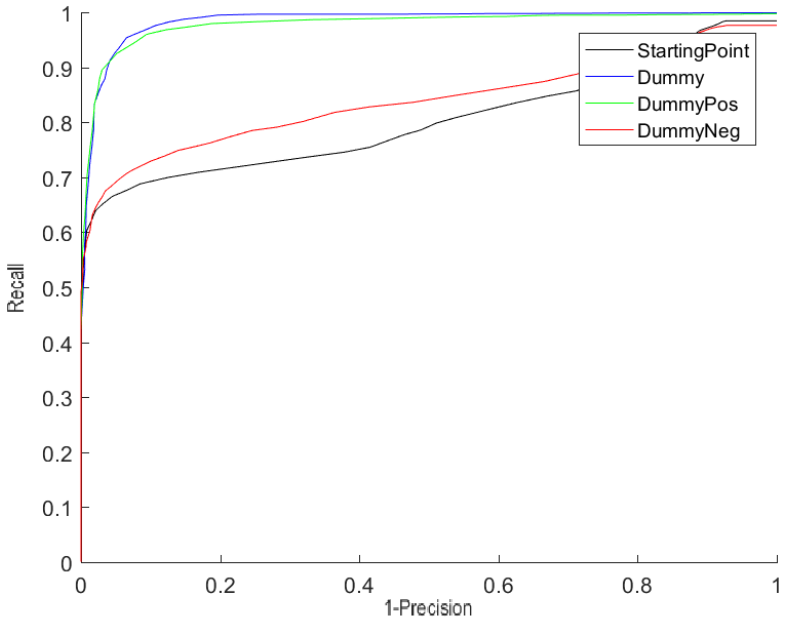

Figura A.1: Curva Precisión-Recall sobre la secuencia 'PETS2006'

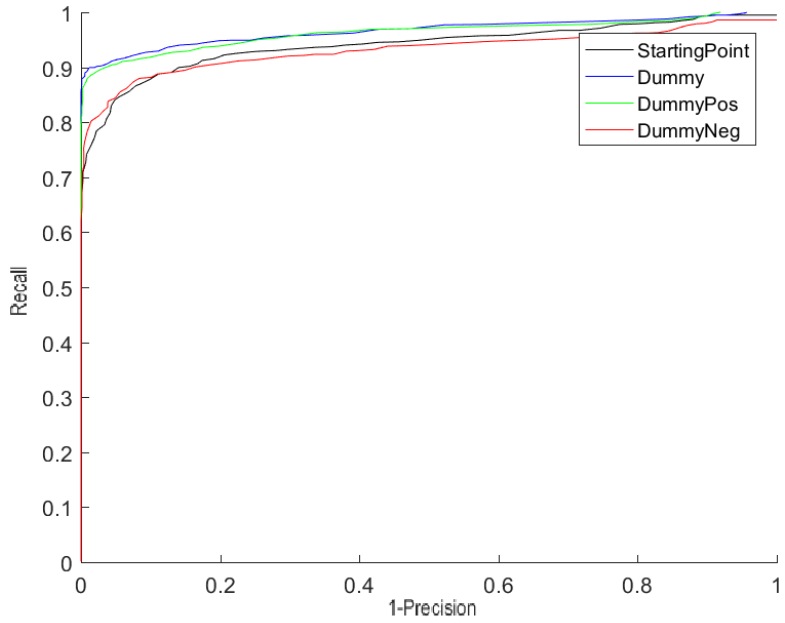

Figura A.2: Curva Precisión-Recall sobre la secuencia 'backdoor'

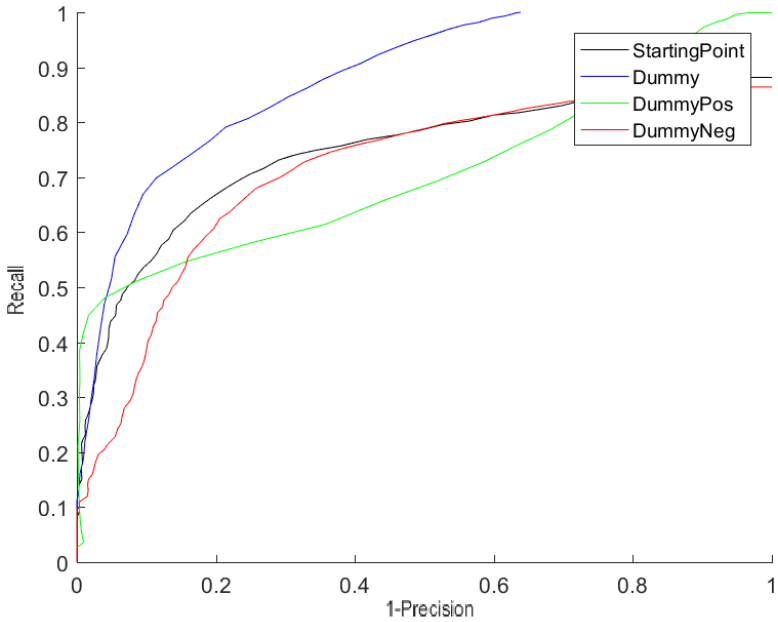

Figura A.3: Curva Precisión-Recall sobre la secuencia 'badminton'

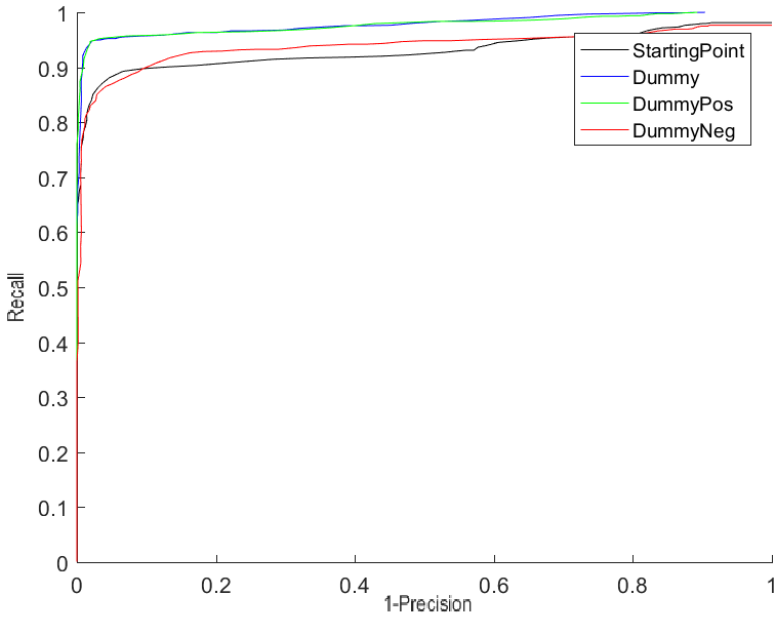

Figura A.4: Curva Precisión-Recall sobre la secuencia 'busStation'

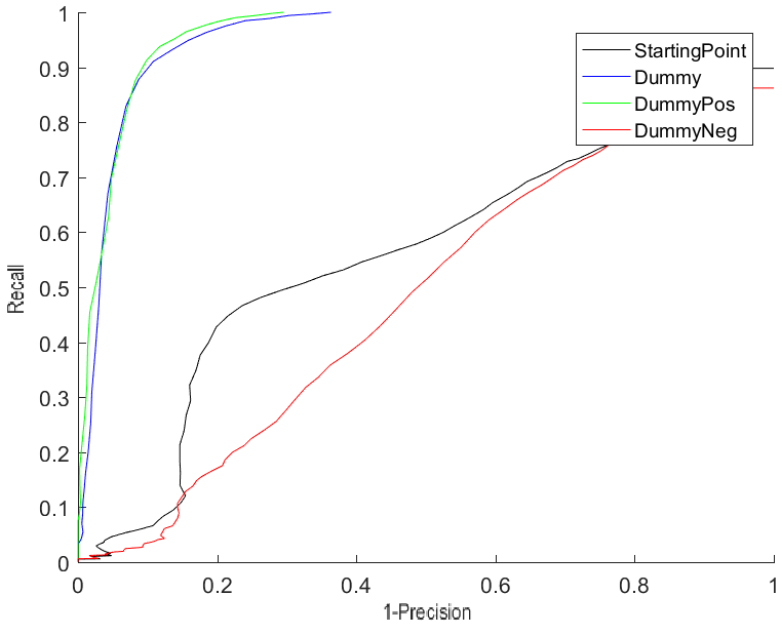

Figura A.5: Curva Precisión-Recall sobre la secuencia 'copyMachine'

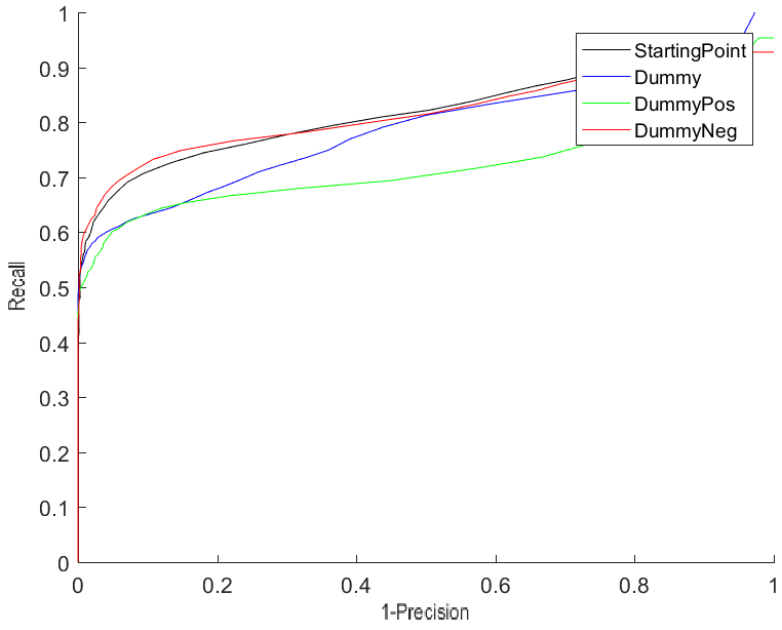

Figura A.6: Curva Precisión-Recall sobre la secuencia 'cubicle'

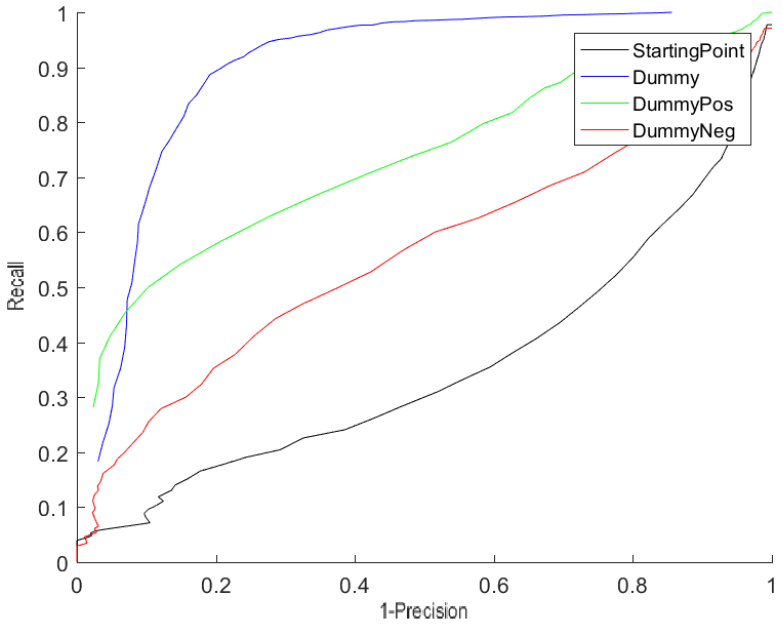

Figura A.7: Curva Precisión-Recall sobre la secuencia 'fall'

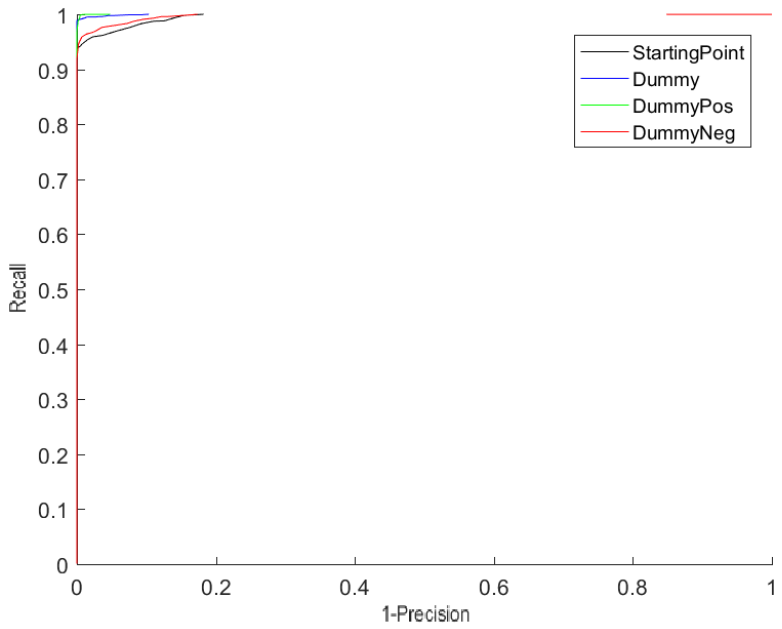

Figura A.8: Curva Precisión-Recall sobre la secuencia 'office'

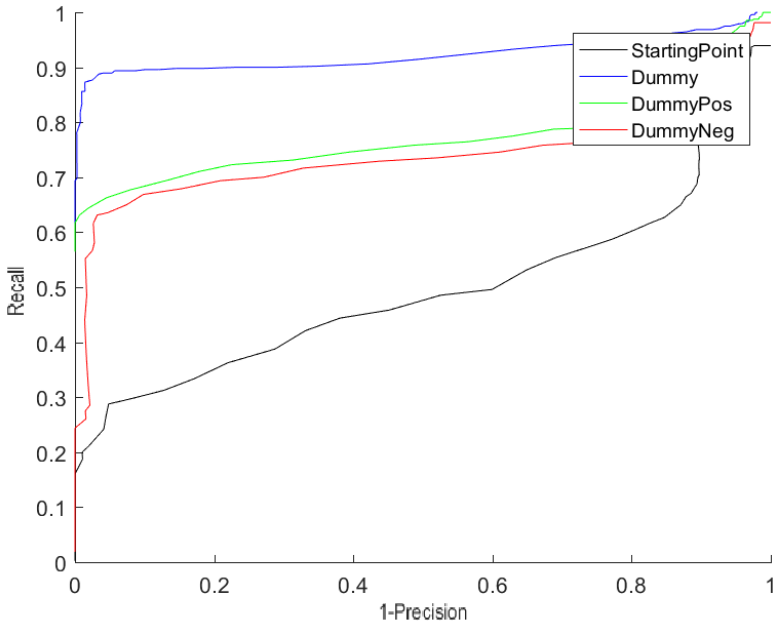

Figura A.9: Curva Precisión-Recall sobre la secuencia 'overpass'

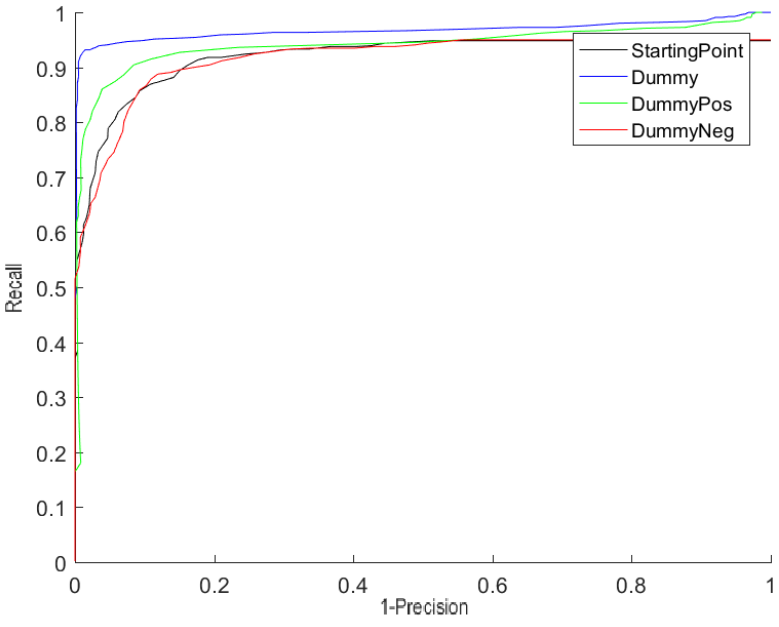

Figura A.10: Curva Precisión-Recall sobre la secuencia 'pedestrians'

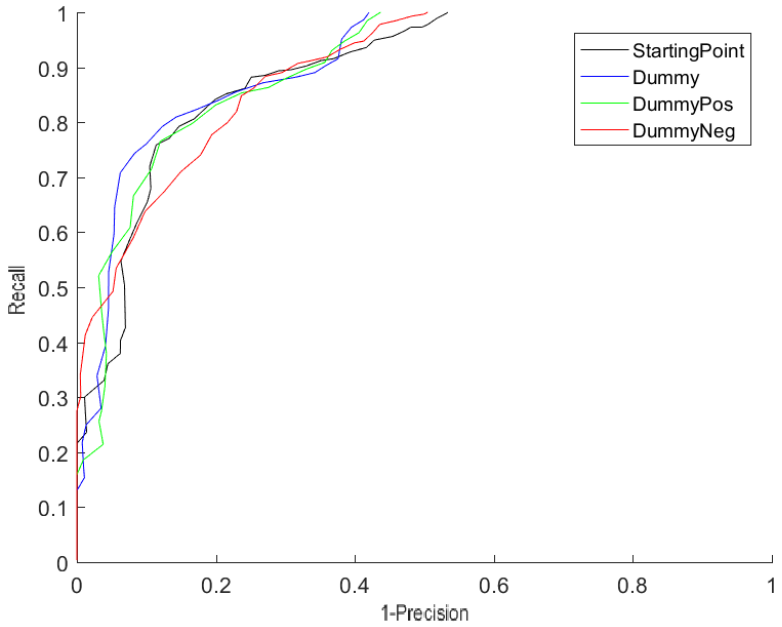

Figura A.11: Curva Precisión-Recall sobre la secuencia 'peopleInShade'

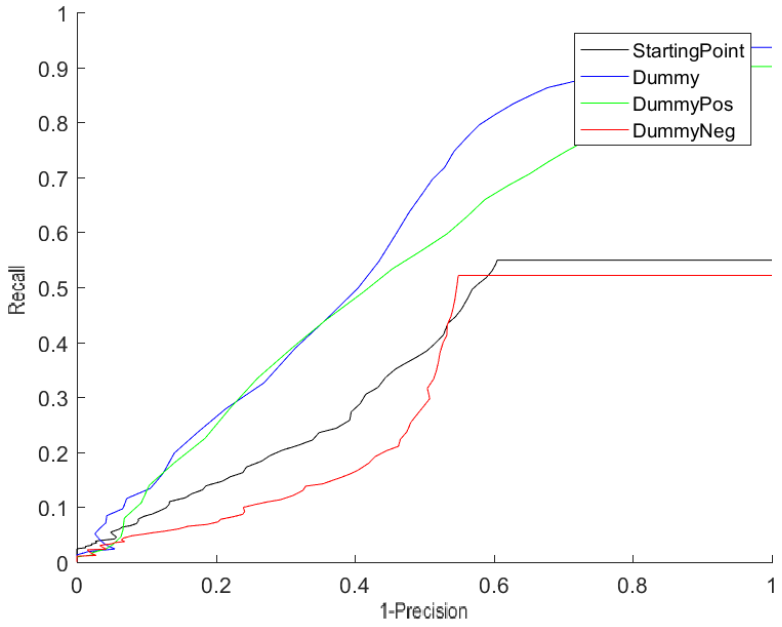

Figura A.12: Curva Precisión-Recall sobre la secuencia 'sidewalk'

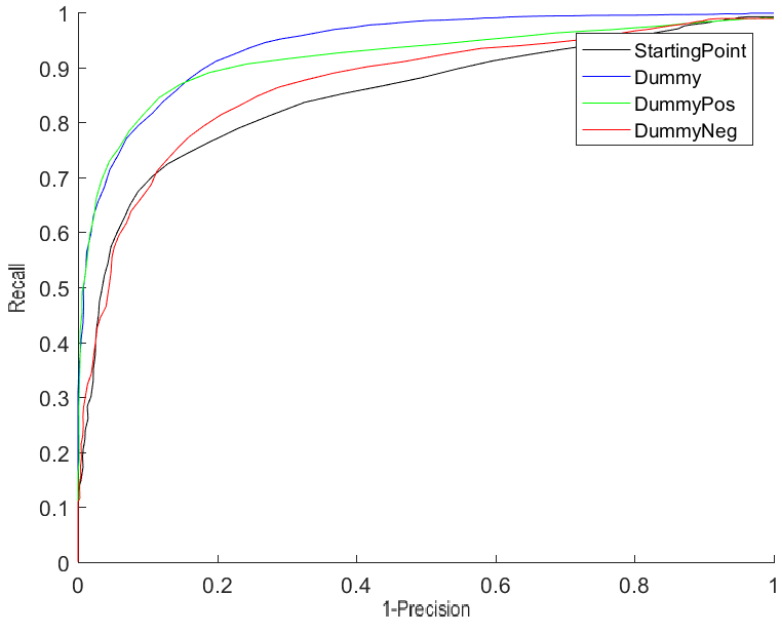

Figura A.13: Curva Precisión-Recall sobre la secuencia 'skating'

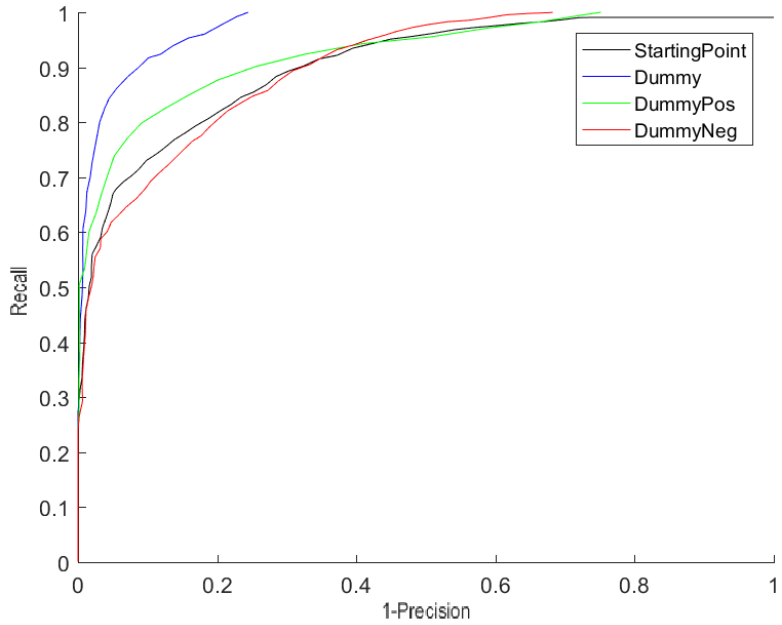

Figura A.14: Curva Precisión-Recall sobre la secuencia 'sofa'

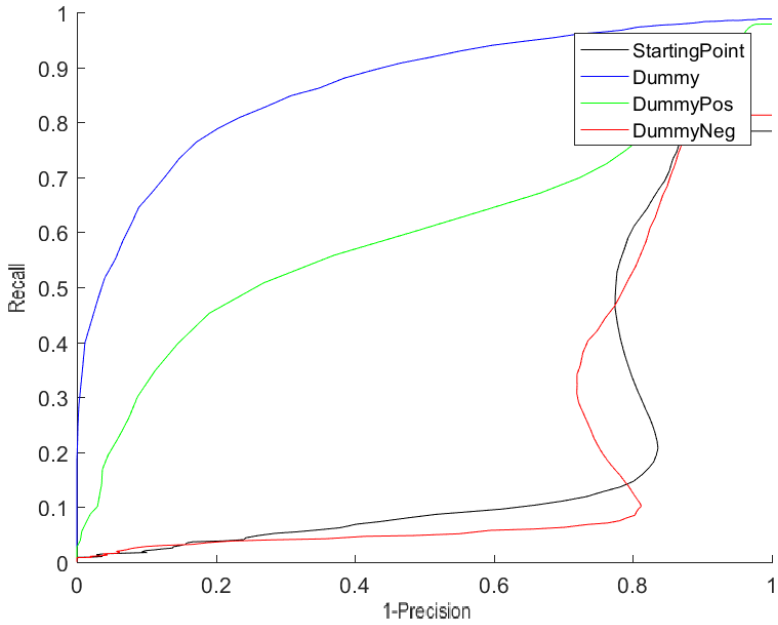

Figura A.15: Curva Precisión-Recall sobre la secuencia 'tramstop'

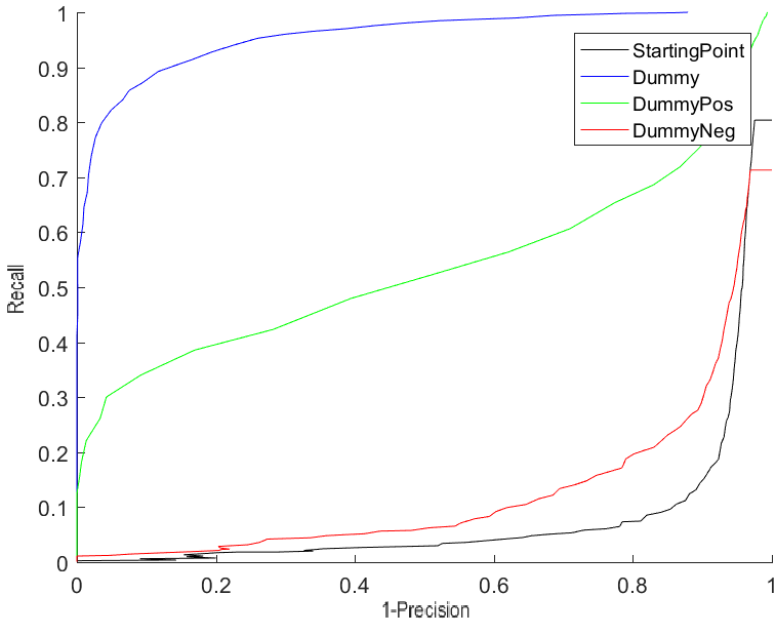

Figura A.16: Curva Precisión-Recall sobre la secuencia 'wetSnow'

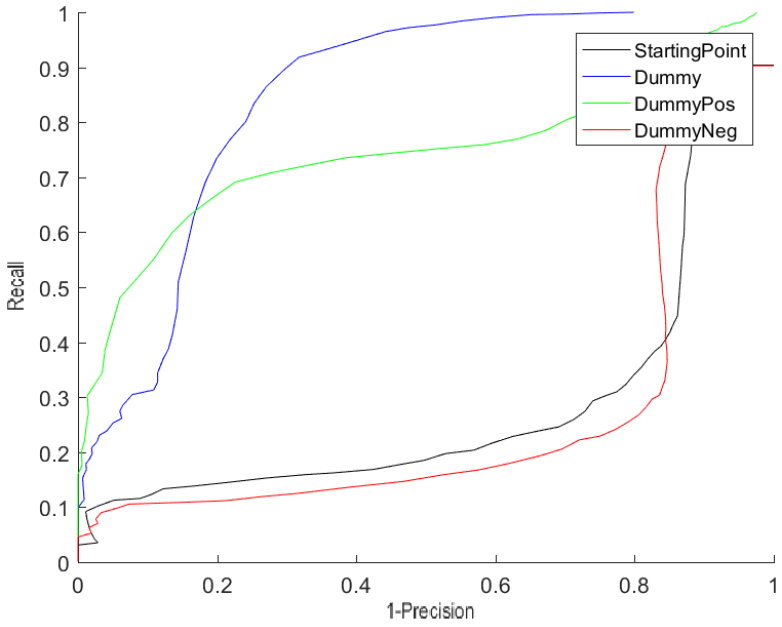

Figura A.17: Curva Precisión-Recall sobre la secuencia 'winterDriveway'

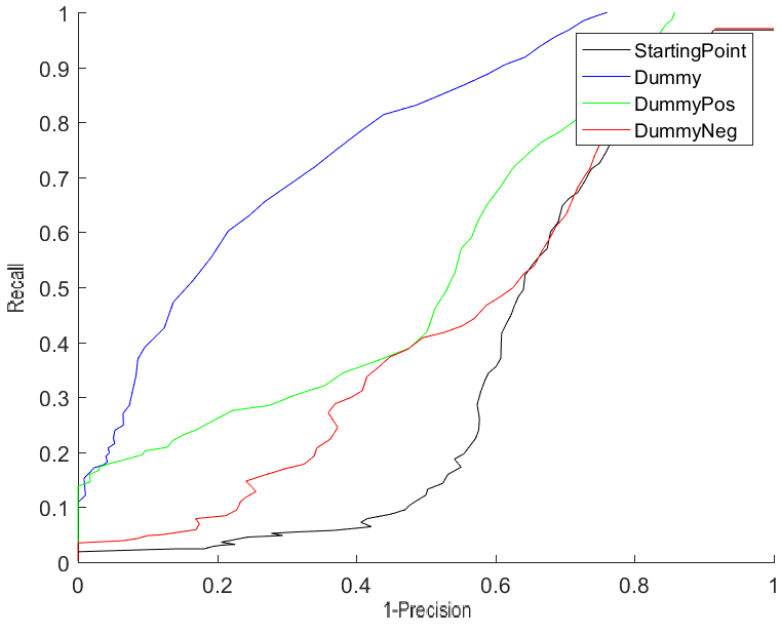

Figura A.18: Curva Precisión-Recall sobre la secuencia 'zoomInZoomOut'

Apéndice B

Resultados del algoritmo propuesto

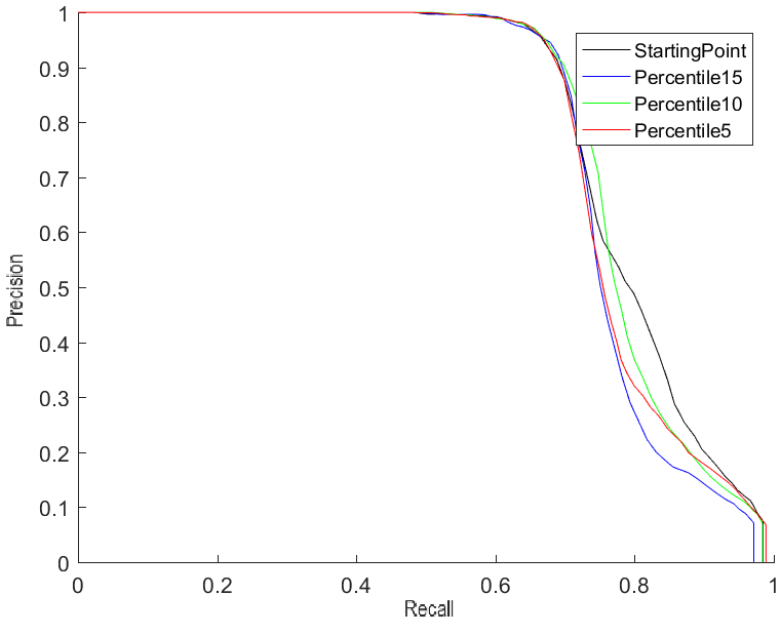

Figura B.1: Curva Precisión-Recall sobre la secuencia 'PETS2006'

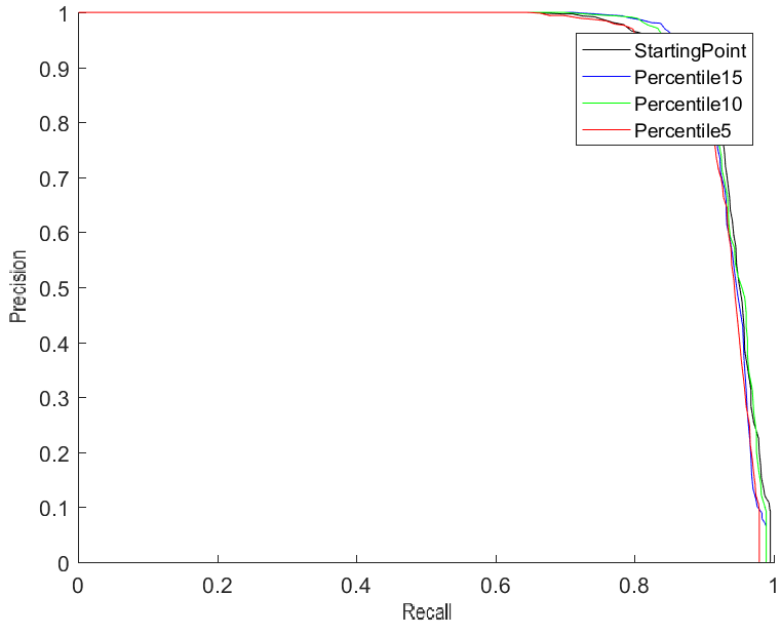

Figura B.2: Curva Precisión-Recall sobre la secuencia 'backdoor'

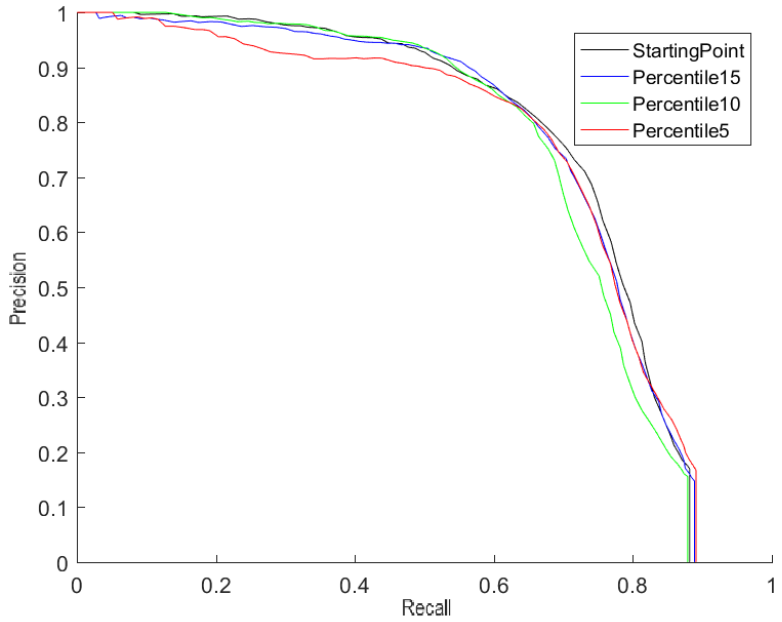

Figura B.3: Curva Precisión-Recall sobre la secuencia 'badminton'

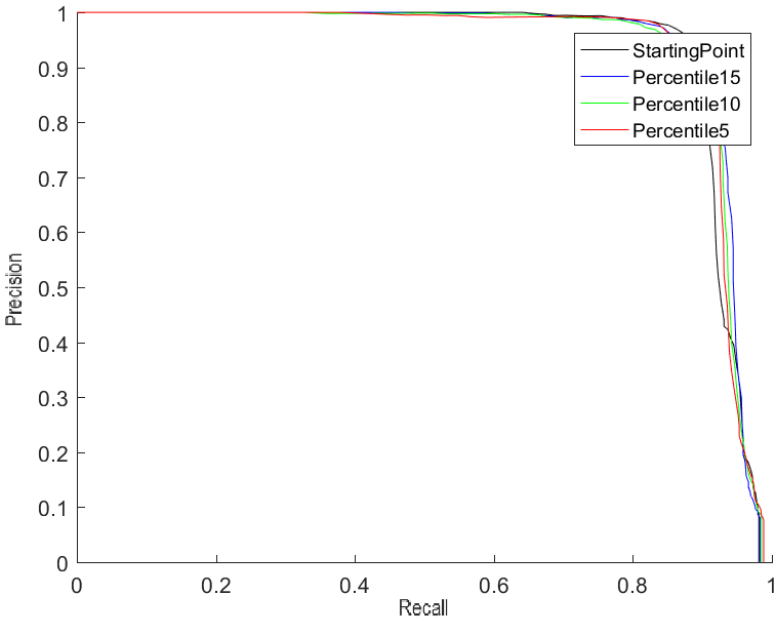

Figura B.4: Curva Precisión-Recall sobre la secuencia 'busStation'

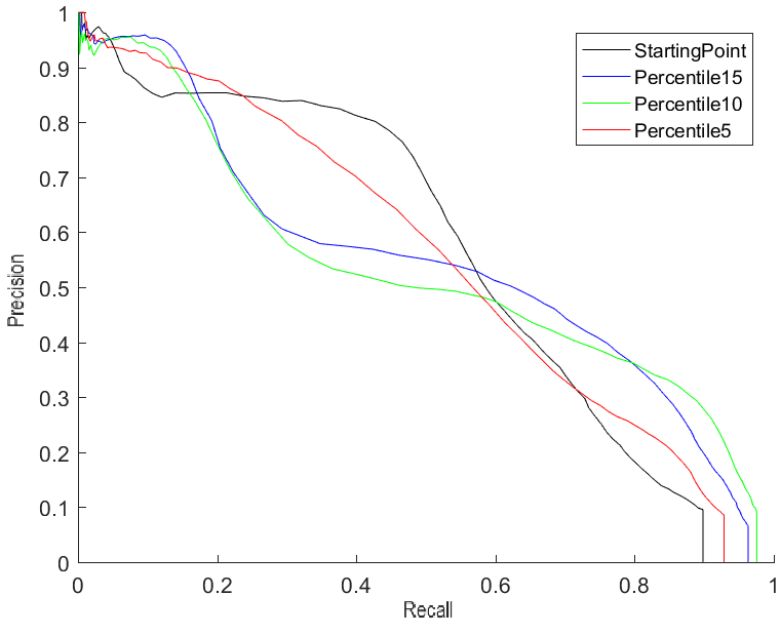

Figura B.5: Curva Precisión-Recall sobre la secuencia 'copyMachine'

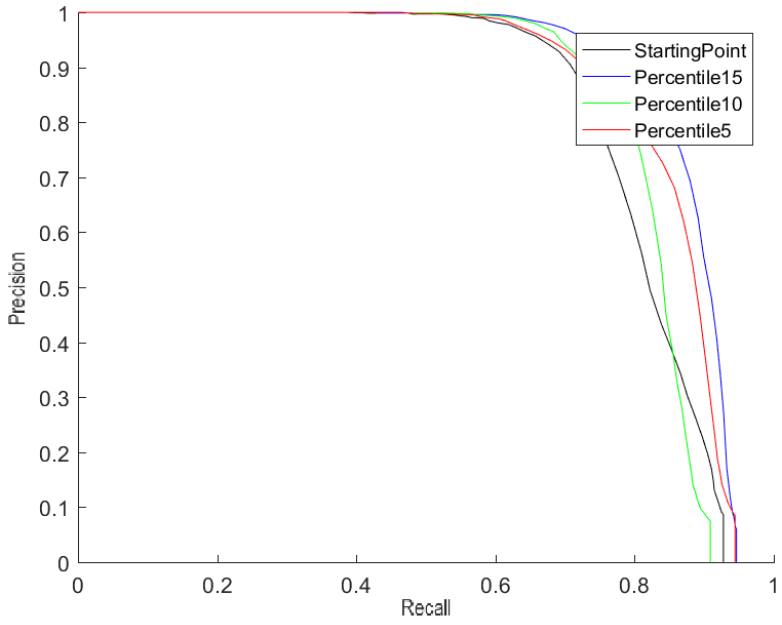

Figura B.6: Curva Precisión-Recall sobre la secuencia 'cubicle'

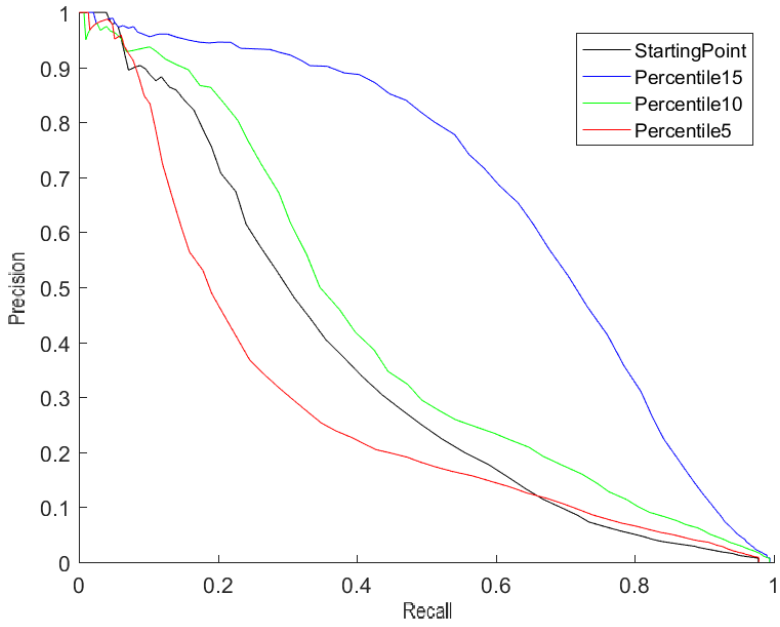

Figura B.7: Curva Precisión-Recall sobre la secuencia 'fall'

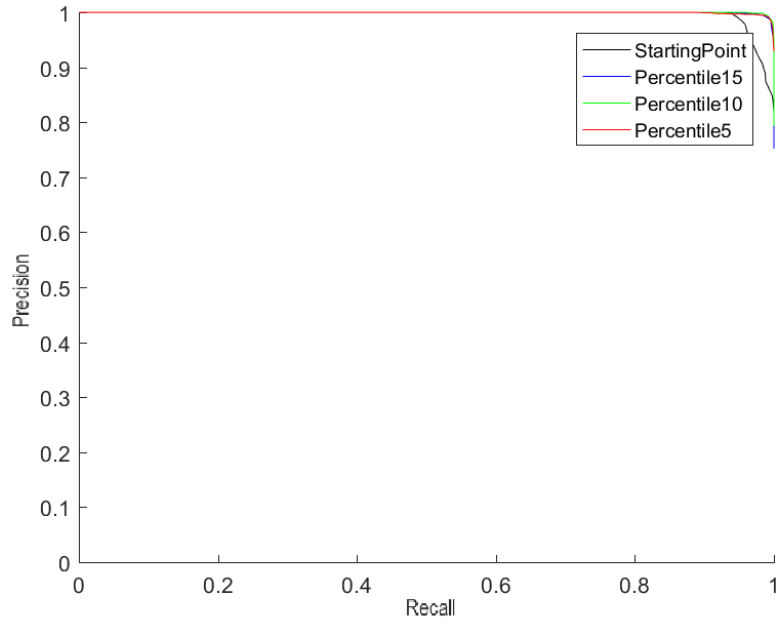

Figura B.8: Curva Precisión-Recall sobre la secuencia 'office'

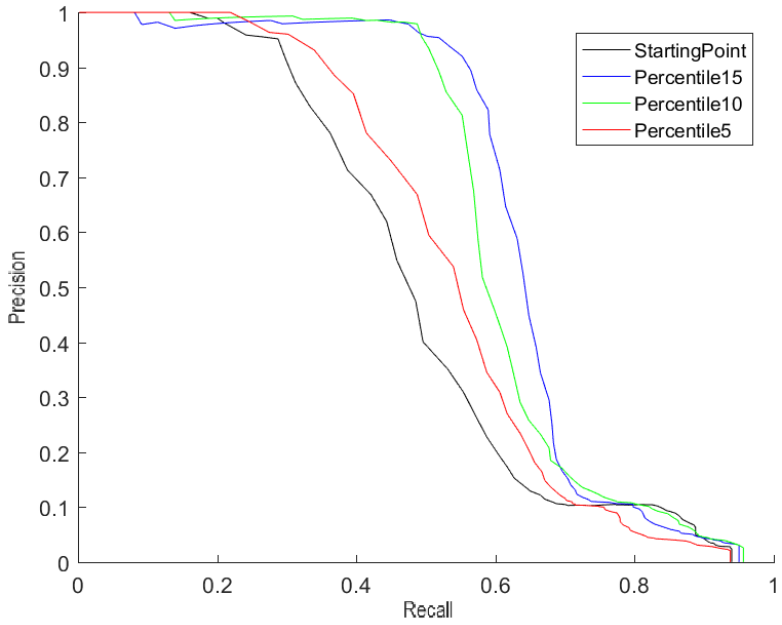

Figura B.9: Curva Precisión-Recall sobre la secuencia 'overpass'

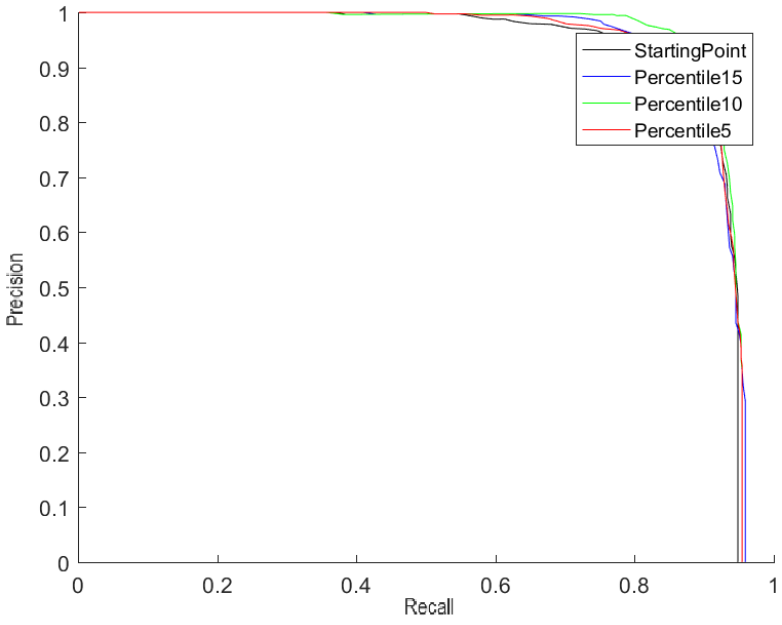

Figura B.10: Curva Precisión-Recall sobre la secuencia 'pedestrians'

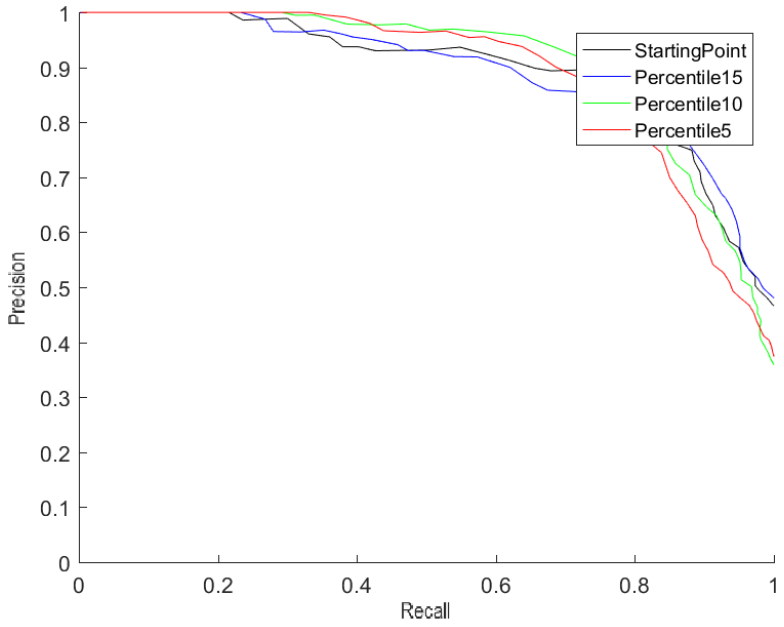

Figura B.11: Curva Precisión-Recall sobre la secuencia 'peopleInShade'

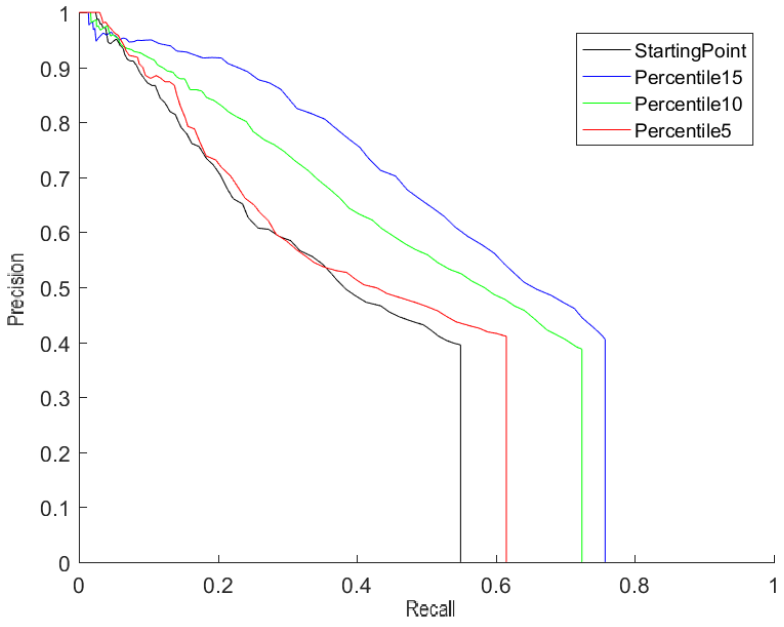

Figura B.12: Curva Precisión-Recall sobre la secuencia 'sidewalk'

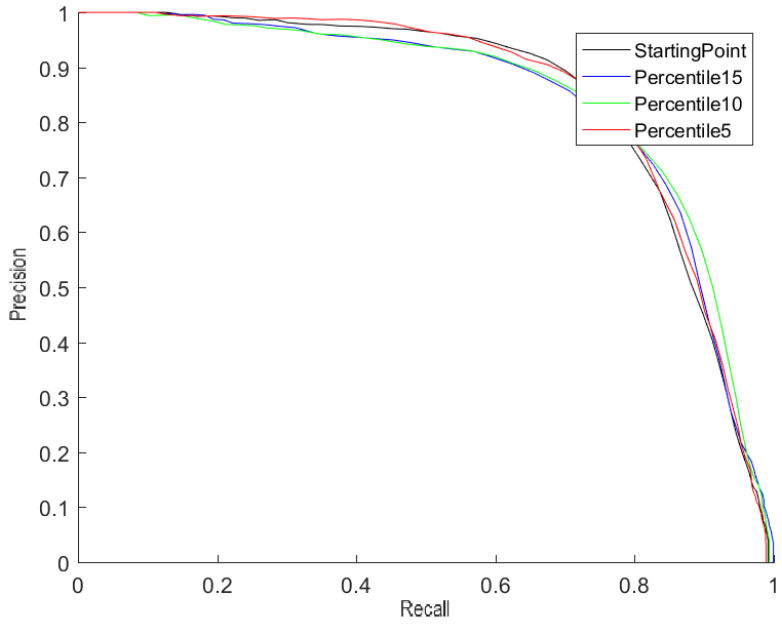

Figura B.13: Curva Precisión-Recall sobre la secuencia 'skating'

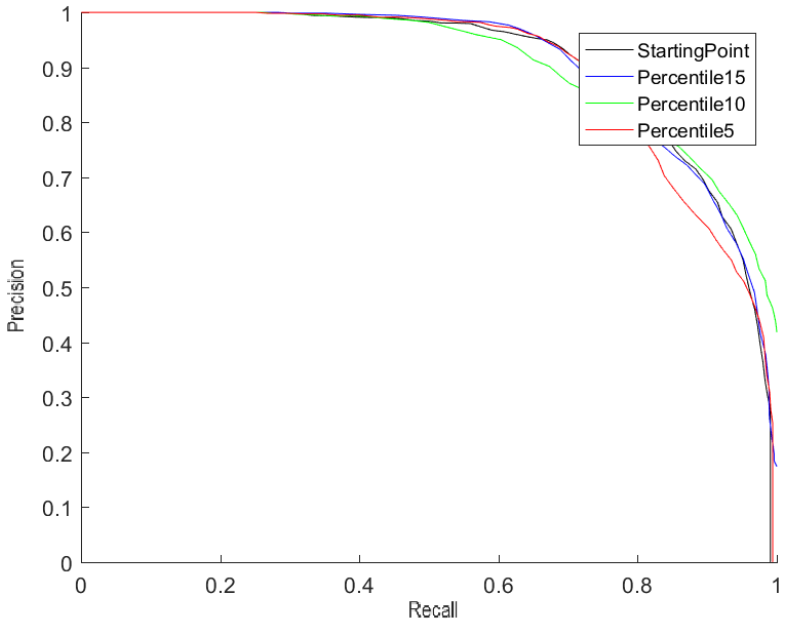

Figura B.14: Curva Precisión-Recall sobre la secuencia 'sofa'

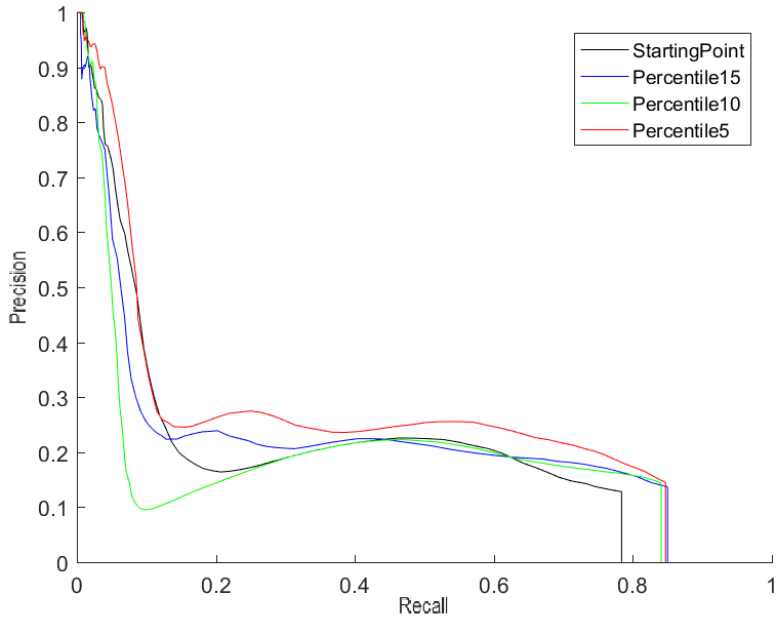

Figura B.15: Curva Precisión-Recall sobre la secuencia 'tramstop'

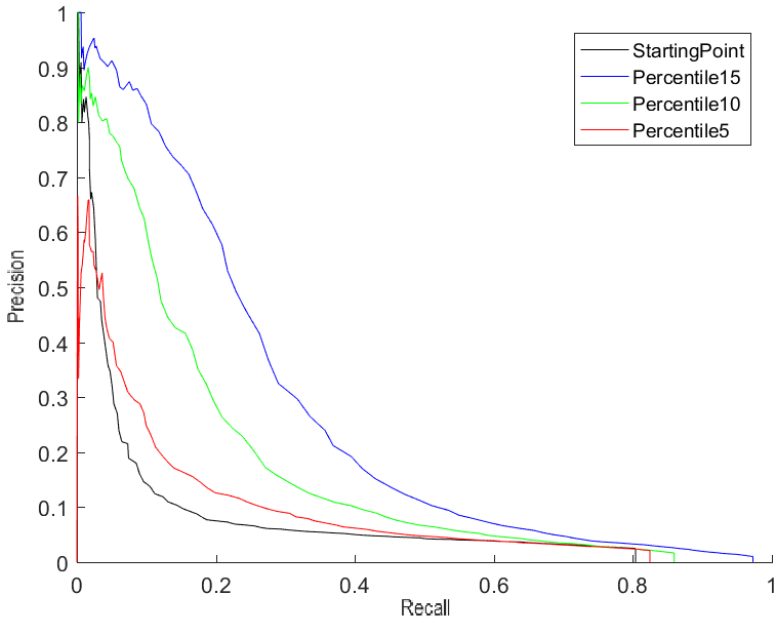

Figura B.16: Curva Precisión-Recall sobre la secuencia 'wetSnow'

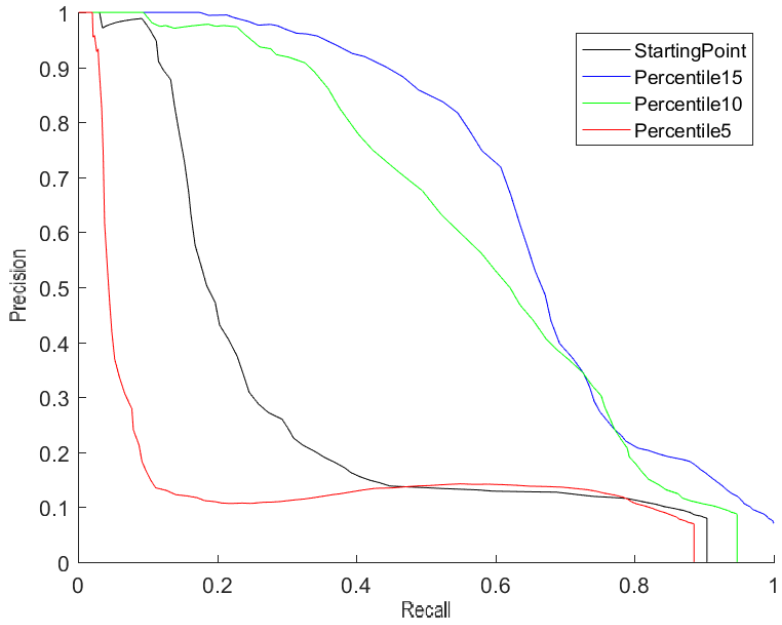

Figura B.17: Curva Precisión-Recall sobre la secuencia 'winterDriveway'

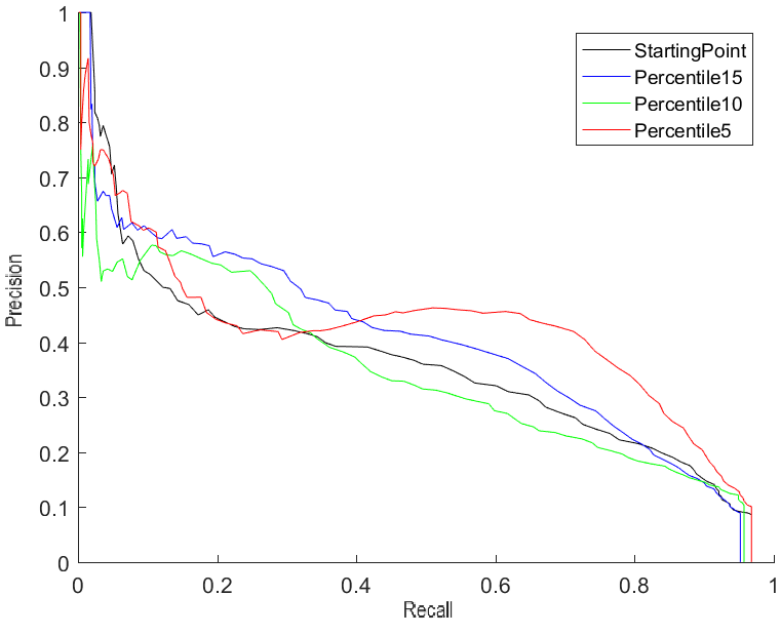

Figura B.18: Curva Precisión-Recall sobre la secuencia 'zoomInZoomOut'